GoldenGate 12c Activity Guide:

Installation & Setup

Oracle Database & GoldenGate Installation and configuration:

Setup Source & Target Database for Replication:

Source DB Name	Target DB Name
Source	Target

On Source:

Install GoldenGate Software

Create GG home directory for GG software installation

[oracle@oracledb /]$ mkdir /u01/app/oracle/product/12.1.0/source

[oracle@oracledb /]$ cd /u01/dump

[oracle@oracledb dump]$ cd GoldenGate/

[oracle@oracledb GoldenGate]$ unzip 121200_fbo_ggs_Linux_x64_shiphome.zip

[oracle@oracledb GoldenGate]$ cd fbo_ggs_Linux_x64_shiphome/Disk1

$ ./runInstaller

Set the following environment for source database after GoldenGate software has been successfully installed.

GG installation Directory for Source (/u01/app/oracle/product/12.1.0/source)

$ vi /u01/app/oracle/product/12.1.0/source/.source

```
export ORACLE_SID=source
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/u01/app/oracle/product/12.1.0/source
export PATH
```

Run .source profile
$ . .source

$ sqlplus / as sysdba

SQL> CREATE TABLESPACE oggtabsp DATAFILE
'/u01/app/oracle/oradata/SOURCE/datafile/oggtabsp.dbf' SIZE 50m AUTOEXTEND ON;
SQL> CREATE USER ggate IDENTIFIED BY oracle123 DEFAULT TABLESPACE oggtabsp;
SQL> GRANT dba TO ggate

Enable minimal supplemental logging by executing the following commands:
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
SQL> ALTER DATABASE FORCE LOGGING;
SQL> ALTER SYSTEM SWITCH LOGFILE
SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;

On Target

Install GoldenGate Software

Create GG home directory for GG software installation
[oracle@oracledb /]$ mkdir /u01/app/oracle/product/12.1.0/target

[oracle@oracledb /]$ cd /u01/dump

[oracle@oracledb dump]$ cd GoldenGate/

[oracle@oracledb GoldenGate]$ unzip 121200_fbo_ggs_Linux_x64_shiphome.zip

[oracle@oracledb GoldenGate]$ cd fbo_ggs_Linux_x64_shiphome/Disk1

$ ./runInstaller

Set the following environment for target database after GoldenGate Software has been successfully installed.
GG installation Directory for Target (/u01/app/oracle/product/12.1.0/target)
$ vi /u01/app/oracle/product/12.1.0/target/.target
export ORACLE_SID=target

export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/u01/app/oracle/product/12.1.0/target
export PATH

Run .target profile

$ . .target

$ sqlplus / as sysdba

SQL> CREATE TABLESPACE oggtabsp DATAFILE '/u01/app/oracle/oradata/TARGET/datafile/oggtabsp.dbf' SIZE 50m AUTOEXTEND ON;
SQL> CREATE USER ggate IDENTIFIED BY oracle123 DEFAULT TABLESPACE oggtabsp;
SQL> GRANT dba TO ggate

Minimal supplemental logging must be enabled at the database level to allow Oracle GoldenGate to properly capture updates to primary keys and chained rows. To enable supplemental logging at the database level, perform the following:

Enable minimal supplemental logging by executing the following commands:
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
SQL> ALTER DATABASE FORCE LOGGING;
SQL> ALTER SYSTEM SWITCH LOGFILE
SQL> SELECT supplemental_log_data_min, force_logging FROM v$database;
Note : An Oracle GoldenGate user requires a database user with at least the following privileges:

User Privilege	**Extract(Source Side)**	**Replicat(Target Side)**
CREATE SESSION, ALTER SESSION	X	X
RESOURCE	X	X
SELECT ANY DICTIONARY	X	X
FLASHBACK ANY TABLE or FLASHBACK ON <owner.table>	X	
SELECT ANY TABLE or SELECT ON <owner.table>	X	X
INSERT, UPDATE, DELETE ON <target tables>		X
CREATE TABLE		X

EXECUTE on DBMS_FLASHBACK package	X	

In addition to the above privileges, you would normally run:

SQL> EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('GGATE');

(Optional) : To learn about the DBMS_GOLDENGATE_AUTH syntax, enter the following commands:

[oracle@oracledb source] $ sqlplus / as sysdba
SQL> set pages 100
SQL> desc dbms_goldengate_auth
SQL> SELECT text FROM **all_source** WHERE name='DBMS_GOLDENGATE_AUTH';

I. Classic Method:

Exercise 1:

Steps for performing an Initial Data Load Method

Note: Make sure to create a same table structure on **Source** and **Target** and Manager Process must be running.

Sample Table Structures (Source & Target)

(Assume Sender & Receiver Schema have already created on Source & Target)

Source:
User : Sender
Table : empl
SQL> Create table sender.empl (empid number (10), empname varchar2 (10), constraint emp_key unique (empid));

Target:
User : Receiver
Table : empl
SQL> Create table receiver.empl (empid number (10), empname varchar2 (10), constraint emp_key unique (empid));

On Source

GGSCI (Source) 8> add extract intdata, sourceistable

EXTRACT added.

GGSCI (Source) 1> edit params **intdata**

```
extract intdata
userid  ggate@source,password ggate
rmthost  192.168.0.35, mgrport 7813
rmttask replicat, group intrep
table sender.empl;
```

On Target

GGSCI> add replicat intrep, specialrun

GGSCI> edit params **intrep**

```
replicat  intrep
userid ggate@target,password ggate
assumetargetdefs
MAP sender.empl, TARGET receiver.empl;
```

Insert some bulk data into source table (sender.empl)

On Source

GGSCI > start extract intdata

Result: Verify Target table has been successfully replicated all data from source table. If not, check logs or report for error

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

Exercise 2:

Steps for performing an Online change Synchronization (CDC) – Without Datapump

Prerequisite:

Assume: Replication tables are already present on both source & target

Create a directory for trail file's location.

$ mkdir /u01/app/oracle/product/12.1.0/target/dirdat/dmltrail

On Target

GGSCI> edit params .GLOBALS
Checkpointtable ggate.chkptab

GGSCI> dblogin userid ggate,password ggate
GGSCI> add checkpointtable ggate.chkptab

On Source

GGSCI> add extract occext, tranlog, Begin now

GGSCI> edit params occext

extract occext
userid ggate@source,password ggate
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/dmltrail/rt
table sender.empl;

GGSCI> add rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/dmltrail/rt, extract occext

On Target

GGSCI> add replicat occrep, exttrail /u01/app/oracle/product/12.1.0/target/dirdat/dmltrail/rt, checkpointtable ggate.chkptab

GGSCI> edit params occrep
replicat occrep
userid ggate@target,password ggate
assumetargetdefs
MAP sender.empl, TARGET receiver.empl;

On Source & Target

GGSCI > start extract occext

GGSCI> start replicat occrep

Do some insert and update on source table which should be reflected on target automatically.

Result: Verify Target table has been successfully replicated all changes at source table. If not, check logs or report for error

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

Exercise 3:

Steps for performing an Online change Synchronization through Data pump process

Prerequisite:

Assume: Replication tables are already present on both source & target

Create a directory for trail file's location.

$ mkdir /u01/app/oracle/product/12.1.0/**source**/dirdat/dptrail
$ mkdir /u01/app/oracle/product/12.1.0/**target**/dirdat/dptrail

On Source

Primary Extract (etsource)

GGSCI> add extract etsource, tranlog, Begin now

GGSCI> edit params **etsource**

```
extract etsource
userid ggate@source,password ggate
exttrail /u01/app/oracle/product/12.1.0/source/dirdat/dptrail/lt
table sender.empl;
```

GGSCI> add exttrail /u01/app/oracle/product/12.1.0/source/dirdat/dptrail/lt, extract etsource

Secondary Extract(etpump) – for pump

GGSCI> add extract etpump, exttrailsource /u01/app/oracle/product/12.1.0/source/dirdat/dptrail/lt

GGSCI> edit params **etpump**

```
Extract etpump
userid ggate@source,password ggate
rmthost 192.168.0.35,mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/dptrail/rt
passthru
table sender.empl;
```

GGSCI> add rmttrail /u01/app/oracle/product/11.2.0/GG2/dirdat/dptrail/rt,extract etpump

On Target

GGSCI> add replicat pumprep, exttrail /u01/app/oracle/product/12.1.0/target/dirdat/dptrail/rt,checkpointtable ggate.chkptab

GGSCI> edit params pumprep

```
replicat pumprep
userid ggate@target,password ggate
assumetargetdefs
MAP sender.empl, TARGET receiver.empl;
```

On Source

GGSCI > start extract etsource
GGSCI> start extract etpump

On Target

GGSCI> start replicat pumprep

Do some insert and update on source table which should be reflected on target automatically.

Result: Verify Target table has been successfully replicated all changes at source table. If not, check logs or report for error

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

Exercise 4:

Steps for Performing Schema Replication through Datapump (DDL – Replication)

Prerequisite:

Assume: Replication schema and tables are already present on both source & target

Create a directory for trail file's location.
$ mkdir /u01/app/oracle/product/12.1.0/**source**/dirdat/schtrail
$ mkdir /u01/app/oracle/product/12.1.0/**target**/dirdat/schtrail

On Source

Primary extract (extsch)

GGSCI> add extract **extsch**, tranlog, begin now

GGSCI> edit params **extsch**
```
extract  extsch
userid ggate@source,password ggate
exttrail /u01/app/oracle/product/12.1.0/source/dirdat/schtrail/lt
ddl include mapped
 table sender.*;
```

GGSCI> add exttrail /u01/app/oracle/product/12.1.0/source/dirdat/schtrail/lt, extract **extsch**

Secondary extract (extpsch)

GGSCI> add extract **extpsch**, exttrailsource /u01/app/oracle/product/12.1.0/source/dirdat/schtrail/lt

GGSCI> edit params **extpsch**
```
extract extpsch
userid ggate@source,password ggate
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/schtrail/rt
passthru
table sender.*;
```

GGSCI> add rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/schtrail/rt, extract **extpsch**

On Target

GGSCI> add replicat **schrep**, exttrail /u01/app/oracle/product/12.1.0/target/dirdat/schtrail/rt,checkpointtable ggate.chkptab

GGSCI> edit params **schrep**

```
replicat schrep
userid ggate@target,password ggate
assumetargetdefs
MAP sender.*, TARGET receiver.*;
```

On Source

GGSCI > start extract extsch
GGSCI> start extract extpsch

On Target

GGSCI> start replicat schrep

Do some insert and update on both the source table which should be reflected on target tables automatically.

Result: Verify Target table has been successfully replicated all changes at source table. If not, check logs or report for error

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

Exercise 5:

Steps for performing a filtering using change capture.

Prerequisite:

Assume: Replication schema and tables are already present on both source & target

Create a directory for trail file's location.
$ mkdir /u01/app/oracle/product/12.1.0/**target**/dirdat/filtrail

On Source

GGSCI> add extract filter, tranlog, begin now

GGSCI> edit params filter

```
extract filter
userid ggate@source,password ggate
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/filtrail/rt
table sender.empl, FILTER ( @STRFIND ( empname, " JOHN") > 0);
```

GGSCI> add rmttrail /u01/app/oracle/product/12.1.0/**target**/dirdat/filtrail/rt, extract filter

On Target

GGSCI> add replicat filrep, exttrail rmttrail /u01/app/oracle/product/12.1.0/**target**/dirdat/filtrail/rt,checkpointtable ggate.chkptab

GGSCI> edit params filrep

```
replicat filrep
Userid ggate@target,password ggate
assumetargetdefs
MAP sender.empl, TARGET receiver.empl;
```

On Source & Target

GGSCI > start extract filter
GGSCI> start replicat filrep

Do some insert and update on source table which should be reflected on target automatically.

Result: Verify Target table has been successfully replicated all changes at source table. If not, check logs or report for error

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

Exercise 6:

Steps for performing a table differ in column structure – Heterogeneous Method

Sample Table Structures (Source & Target)

Source:
SQL> Alter table sender.empl add (salary number (10, 2));

Target:
SQL> Alter table receiver.empl (sal number (10, 2);

Prerequisite:
Create a directory for trail file's location.

$ mkdir /u01/app/oracle/product/12.1.0/**target**/dirdat/coltrail

On Source

Create definition file on the source by using **defgen** utility, and then copy that definitions file to the target system.

GGSCI > edit params defgen
Defsfile /u01/app/oracle/product/12.1.0/**source**/dirsql/myteam.sql
userid ggate@source,password ggate
table **sender.empl;**

$. /defgen paramfile /u01/app/oracle/product/12.1.0/**source**/dirprm/defgen.prm

Copy **myteam.sql** file to target system location
/u01/app/oracle/product/12.1.0/**target**/dirsql/myteam.sql

GGSCI> add extract **extcol,** tranlog, Begin now

GGSCI> edit params extcol
Extract extcol
userid ggate@source,password ggate
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/**target**/dirdat/coltrail/rt

table **sender.empl;**

On Target

GGSCI> add replicat **repcol**, exttrail /u01/app/oracle/product/12.1.0/**target**/dirdat/coltrail/rt,checkpointtable ggate.chkptab

GGSCI> edit params repcol
replicat repcol
sourcedefs /u01/app/oracle/product/12.1.0/**target**/dirsql/myteam.sql
userid ggate@target,password ggate
MAP sender.empl, TARGET receiver.empl, COLMAP (usedefaults, salary=sal);

On Source & Target

GGSCI > start extract extcol
GGSCI> start replicat repcol

Do some insert and update on source table which should be reflected on target automatically.

Result: Verify Target table has been successfully replicated all changes at source table. If not, check logs or report for error

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

II. Integrated Method

Exercise 6:

Steps for performing integrated Capture and Replicat setup

Database setup for Integrated Capture (Source)

- EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE (grantee => 'ggate', privilege_type => 'capture', grant_select_privileges=> true, do_grants => TRUE);

- When we add the extract we have to use the **INTEGRATED CAPTURE** clause in the ADD EXTRACT command as shown below

GGSCI> ADD EXTRACT intext INTEGRATED TRANLOG, BEGIN NOW

In the extract parameter file we have to use **TRANLOGOPTIONS INTEGRATEDPARAMS** parameter as show below

TRANLOGOPTIONS INTEGRATEDPARAMS (max_sga_size 200, parallelism 1)

(The max_sga_size is denoted in MB and this memory is taken from the streans_pool_size part of the SGA memory. If the streams_pool_size is greater than 1 GB, max_sga_size defaults to 1 GB, otherwise it is 75% of the streans_pool_size)

The parallelism specifies the number of processes supporting the database log mining server. It defaults to 2

Register the extract

We use the REGISTER EXTRACT command to register the primary extract group with the Oracle database. The extract process does not directly read the redo log files as in the classic capture mode, but integrates with the database log mining server to receive changes in the form of Logical Change Records or LCR's.

We do this before adding the extract and must connect to the database first via the DBLOGIN command

SQL> alter system set enable_goldengate_replicate=true scope=both

GGSCI> DBLOGIN USER ggate PASSWORD ggate
GGSCI> REGISTER EXTRACT ext1 DATABASE

Example

GGSCI > DBLOGIN USERID ggate, PASSWORD ggate
Successfully logged into database

GGSCI > REGISTER EXTRACT ext1 DATABASE

GGSCI > ADD EXTRACT ext1 INTEGRATED TRANLOG, BEGIN NOW

EXTRACT added.

GGSCI > ADD RMTTRAIL /u01/app/oracle/product/12.1.0/target/dirdat/rt, EXTRACT ext1
EXTTRAIL added.

GGSCI > EDIT PARAMS ext1
EXTRACT ext1
USERID ggate, PASSWORD ggate
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 100)
EXTTRAIL /u01/app/oracle/product/12.1.0/target/dirdat/rt
TABLE sender.empl;
In the background ….

When we register the extract, we will see that a capture process called OGG$CAP_EXT1 was created and a queue called OGG$Q_EXT1 was created in the GGATE schema.

A good source of information is also the database alert log and we can see messages like the ones shown below:

LOGMINER: session#=1 (OGG$CAP_EXT1), reader MS00 pid=41 OS id=32201 sid=153 started
Thu Jan 24 18:04:15 2013
LOGMINER: session#=1 (OGG$CAP_EXT1), builder MS01 pid=42 OS id=32203 sid=30 started
Thu Jan 24 18:04:15 2013
LOGMINER: session#=1 (OGG$CAP_EXT1), preparer MS02 pid=43 OS id=32205 sid=155 started
Thu Jan 24 18:04:16 2013

LOGMINER: Begin mining logfile for session 1 thread 1 sequence 12, …
LOGMINER: End mining logfile for session 1 thread 1 sequence 12, /u01/

Read further

GoldenGate Integrated Capture Healthcheck Script [Article ID 1448324.1]

Integrate Replicat:

(On the target, add the Integrated Replicat)

GGSCI > DBLOGIN USERID ggate, PASSWORD ggate

GGSCI > ADD REPLICAT rep1 INTEGRATED EXTTRAIL /u01/app/oracle/product/12.1.0/target/dirdat/rt

GGSCI > edit params rep1

REPLICAT rep1

SETENV (ORACLE_SID='target')

DBOPTIONS INTEGRATEDPARAMS (parallelism 6)

USERID ggate, PASSWORD ggate

ASSUMETARGETDEFS

MAP sender.empl, TARGET receiver.empl;

Note:

The parameter DBOPTIONS INTEGRATEDPARAMS (parallelism 6) denotes that this for this integrated replicat, we are specifying that the minimum number of parallel apply processes will be 6.

GGSCI > start rep1

GGSCI > **info replicat rep1**

SQL> select REPLICAT_NAME, SERVER_NAME from DBA_GOLDENGATE_INBOUND;

```
REPLICAT_NAME      SERVER_NAME
------------------ ------------------
MYREP1             OGG$REP1
```

SQL> select APPLY_NAME, QUEUE_NAME, status from dba_apply;

```
APPLY_NAME         QUEUE_NAME         STATUS
------------------ ------------------ --------
OGG$MYREP1         OGGQ$REP1          ENABLED
```

SQL> select apply_name, state from V$GG_APPLY_COORDINATOR;

```
APPLY_NAME                 STATE
-------------------------- --------------------
OGG$MYREP1                 IDLE
```

Note: Because we had configured PARALLELISM to be 6 via the DBOPTIONS INTEGRATEDPARAMS (parallelism 6) in the replicat parameter file, we will see 6 apply server processes which are ready to run.

At this stage they are IDLE and have not received or applied any messages or LCRs.

```
SQL> select server_id, TOTAL_MESSAGES_APPLIED from V$GG_APPLY_SERVER
  2 where apply_name= OGG$REP1';

 SERVER_ID TOTAL_MESSAGES_APPLIED
---------- ---------------------
         4                     0
         2                     0
         6                     0
         5                     0
         3                     0
         1                     0

6 rows selected.
```

Populate the base table and monitor the extract process

We now insert a million rows into our target table and see that the extract has processed those newly added rows.

```
GGSCI > stats extract myext1

Sending STATS request to EXTRACT EXT1 ...

Start of Statistics at 2014-01-20 16:21:26.

DDL replication statistics (for all trails):

*** Total statistics since extract started     ***
    Operations                                  1.00

Output to ./dirdat/ex:

Extracting from SALES.SH.MYTAB to SALES_DR.SH.MYTAB:

*** Total statistics since 2014-01-20 16:19:54 ***
    Total inserts                         1000000.00
    Total updates                               0.00
```

Total deletes 0.00
Total discards 0.00
Total operations 1000000.00

Monitor the Integrated Replicat

GGSCI (orasql-001-dev.mydomain) 2> **info replicat rep1**

REPLICAT REP1 Last Started 2014-01-20 16:12 Status RUNNING
INTEGRATED
Checkpoint Lag 00:00:00 (updated 00:00:06 ago)
Process ID 4794
Log Read Checkpoint File ./dirdat/rx000000
2014-01-20 16:22:51.183273 RBA 54918866

GGSCI > stats replicat rep1

Sending STATS request to REPLICAT REP1 ...

Start of Statistics at 2014-01-20 17:47:25.

Integrated Replicat Statistics:

Total transactions 5.00
Redirected 0.00
DDL operations 0.00
Stored procedures 0.00
Datatype functionality 0.00
Event actions 0.00
Direct transactions ratio 0.00%

Replicating from SENDER.EMPL to RECEIVER.EMPL:

*** Total statistics since 2014-01-20 16:20:05 ***
Total inserts 1000000.00
Total updates 0.00
Total deletes 0.00
Total discards 0.00
Total operations 1000000.00

Monitor the status of the Database Apply Server Processes

SQL> select apply_name, state from V$GG_APPLY_COORDINATOR;

```
APPLY_NAME                     STATE
------------------------------ --------------------
OGG$REP1                       APPLYING


SQL> select server_id, TOTAL_MESSAGES_APPLIED from V$GG_APPLY_SERVER
  2 where apply_name='OGG$REP1';


 SERVER_ID TOTAL_MESSAGES_APPLIED
---------- ----------------------
         4                      0
         2                 388462
         6                      0
         5                      0
         3                      0
         1                 611543


SQL> select apply_name, state, TOTAL_MESSAGES_DEQUEUED,
TOTAL_MESSAGES_SPILLED from V$GG_APPLY_READER;

APPLY_NAME                     STATE
------------------------------ ------------------------------------
TOTAL_MESSAGES_DEQUEUED TOTAL_MESSAGES_SPILLED
----------------------- ----------------------
OGG$REP1                       IDLE
                 1000005                    0


SQL> select APPLY_NAME, TOTAL_APPLIED, TOTAL_RECEIVED from
V$GG_APPLY_COORDINATOR;

APPLY_NAME                     TOTAL_APPLIED TOTAL_RECEIVED
------------------------------ ------------- --------------
OGG$REP1                                   5              5


SQL> select apply_name, state from V$GG_APPLY_COORDINATOR;

APPLY_NAME                     STATE
------------------------------ --------------------

OGG$REP1                       IDLE
```

Exercise 7:

Steps for performing Online Change Synchronization with the initial data load – Handlecollisions

How do we manage/handle changes that are happening to the data while the **initial data load extract process** is in operation? Sometimes it may not be possible to have an application outage just to peform an initial data load and in most cases we will need to perform the initial data load using GoldenGate while users are connected to the database and changes are being made to the database via the application.

Example, we will be performing an initial data load of the TEST_OBJ table (copy of DBA_OBJECTS) and while the initial data load extract process is running and loading the 90,000 rows, we will from another session update the table while the data load is in progress. We will then see how these changes are also replicated to the target

On Source

```
SQL> grant dba to sender;
Grant succeeded.

SQL> connect sender/sender;
Connected.
SQL> create table test_obj as select * from sys.dba_objects;

SQL> select count(*) from test_obj;

  COUNT(*)
----------
     91072
```

On Target

```
SQL> grant dba to receiver;
Grant succeeded.

SQL> connect receiver/receiver;
Connected.

SQL> create table test_obj as ( select * from dba_objects where 1=2);
Table created.
```

SQL> select count(*) from test_obj;

COUNT(*)

0

For Initial load Process …

On Source

GGSCI (oracledb) 3> add extract **extr1,** sourceistable
EXTRACT added.

GGSCI (oracledb) 4> edit params **extr1**
extract extr1
userid ggate, password ggate
rmthost 192.168.0.35, mgrport 7813
rmttask replicat, GROUP repl1
table sender.test_obj;

On Target

GGSCI (oracledb) 4> add replicat **repl1,** specialrun
REPLICAT added.

GGSCI (oracledb) 5> edit params **repl1**
replicat repl1
HANDLECOLLISIONS
userid ggate,password ggate
assumetargetdefs
MAP sender.test_obj, TARGET receiver.test_obj;

For Online Change Synchronization…

On Source

GGSCI (oracledb) 8> add extract **ext2**, tranlog, begin now
EXTRACT added.

GGSCI (oracledb) 9> add rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/ha
ndle/rt, extract **ext2**
RMTTRAIL added.

GGSCI (oracledb) 1> edit params **ext2**

extract ext2
userid ggate, password ggate
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/handle/rt
TABLE sender.test_obj;

GGSCI (oracledb) 1> dblogin userid ggate, password ggate
Successfully logged into database.

GGSCI (oracledb) 2> add trandata sender.test_obj

On Target

GGSCI (oracledb) 2> add replicat **rep2**, exttrail **/u01/app/oracle/product/12.1.0/target/dirdat/handle/rt**, checkpointtable ggate.chkptab

REPLICAT added.

GGSCI (oracledb) 1> edit params **rep2**
replicat rep2
HANDLECOLLISIONS
ASSUMETARGETDEFS
userid ggate, password ggate
MAP sender.test_obj, TARGET receiver.test_obj;

On Source

Start the Online Change Extract ext2…

GGSCI (oracledb) 2> start extract **ext2**

Sending START request to MANAGER...
EXTRACT EXT2 starting

GGSCI (oracledb) 11> info extract **ext2**

EXTRACT EXT2 Last Started 2016-04-15 05:53 Status RUNNING
Checkpoint Lag 00:00:00 (updated 00:00:02 ago)
Process ID 5491
Log Read Checkpoint Oracle Redo Logs
2016-04-15 05:53:30 Seqno 11, RBA 29640704
SCN 0.1873006 (1873006)

Start the Initial Load Extract extr1…

GGSCI (oracledb) 12> start extract **extr1**

Sending START request to MANAGER ...
EXTRACT EXTR1 starting

GGSCI (oracledb) 13> info extract **extr1**

EXTRACT EXTR1 Initialized 2016-04-15 05:39 Status STARTING
Checkpoint Lag Not Available
Process ID 5519
Log Read Checkpoint Not Available
First Record Record 0
Task SOURCEISTABLE

GGSCI (oracledb) 14> info extract **extr1**

EXTRACT EXTR1 Last Started 2016-04-15 05:54 Status RUNNING
Checkpoint Lag Not Available
Process ID 5519
Log Read Checkpoint Table SENDER.TEST_OBJ
2016-04-15 05:54:33 Record 1
Task SOURCEISTABLE

GGSCI (oracledb) 15> info extract **extr1**

EXTRACT EXTR1 Last Started 2016-04-15 05:54 Status RUNNING
Checkpoint Lag Not Available
Process ID 5519
Log Read Checkpoint Table SENDER.TEST_OBJ
2016-04-15 05:54:43 Record 33624
Task SOURCEISTABLE

While the Initial Load Extract is in progress make some changes in the database

SQL> connect sender/sender
Connected.

SQL> update test_obj set owner=**'SRINI'** where owner=**'SYS';**
41885 rows updated.

SQL> commit;
Commit complete.

When the initial extract process has loaded all the rows, it will stop and so will the initial replicat process

GGSCI (oracledb) 1> info extract **extr1**

EXTRACT EXTR1 Last Started 2016-04-15 05:54 Status STOPPED
Checkpoint Lag Not Available
Log Read Checkpoint Table SENDER.TEST_OBJ
2016-04-15 05:54:51 Record 91072
Task SOURCEISTABLE

On Target

GGSCI (oracledb) 4> send replicat **rep1 getlag**

GGSCI (oracledb) 5> start replicat **rep2**

Sending START request to MANAGER ...
REPLICAT REP2 starting

GGSCI (oracledb) 11> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag **237** seconds.

GGSCI (oracledb) 12> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag **252** seconds.

GGSCI (oracledb) 13> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag **258** seconds.

GGSCI (oracledb) 14> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag **261** seconds.

GGSCI (oracledb) 15> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag **264** seconds.

GGSCI (oracledb) 16> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag **266** seconds.

GGSCI (oracledb) 17> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag 269 seconds.

GGSCI (oracledb) 18> send replicat rep2 getlag

Sending GETLAG request to REPLICAT REP2 ...
Last record lag 272 seconds.
At EOF, no more records to process.

Let us now check if both the initial data load and the updates have been propagated and applied on the target side.

SQL> connect receiver/receiver
Connected.

SQL> select count(*) from test_obj;

```
  COUNT(*)
----------
     91072
```

```
SQL> select count(*) from test_obj where owner='SRINI';

  COUNT(*)
----------
     41885
```

Now remove the HANDLECOLLISIONS clause …

GGSCI (oracledb) 1> send replicat rep2, nohandlecollisions

Sending NOHANDLECOLLISIONS request to REPLICAT REP2 ...
REP2 No tables found matching * to set NOHANDLECOLLISIONS.

Also remove the line from the replicat parameter file via the “edit params replicat rep2 command”

Exercise 8:

Steps for performing - Tokens with COLMAP Clause

Using **@TOKEN** function to extract data stored in token area of GoldenGate Trail file Record Header.

In the example below, the source table has two columns (EMPID and EMPNAME) and the target table has some other columns in addition to these two columns which we will populate using Tokens and the @DATENOW function which will populate the column with the current timestamp.

Prerequisite

Source:

SQL> Create table sender.emptoken (empid number (10), empname varchar2 (10), constraint emp_key unique (empid));

Target:

SQL> Create table receiver.emptoken (empid number (10), empname varchar2 (10), constraint emp_key unique (empid), hostname varchar2(20), osuser varchar2(10), dbname varchar2(10), tran_date date);

On Source

GGSCI (oracledb) 2> add extract **ettoken**, tranlog, begin now

EXTRACT added.

GGSCI (oracledb) 3> edit params **ettoken**

EXTRACT ettoken
USERID ggate, PASSWORD ggate
RMTHOST 192.168.0.35, MGRPORT 7813
RMTTRAIL /u01/app/oracle/product/12.1.0/target/dirdat/ettoken/rt
TABLE sender.emptoken, TOKENS(TK_HOST = @GETENV('GGENVIRONMENT', 'HOSTNAME'), TK_OSUSER=@GETENV('GGENVIRONMENT', 'OSUSERNAME'), TK_DBNAME=@GETENV('DBENVIRONMENT','DBNAME'));

GGSCI (oracledb) 4> add rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/ettoken/rt, extract ettoken

RMTTRAIL added.

On Target

GGSCI (oracledb) 1> add replicat **rptoken**, exttrail /u01/app/oracle/product/12.1.0/target/dirdat/ettoken/rt, checkpointtable ggate.chkptab

REPLICAT added.

GGSCI> edit params rptoken
REPLICAT rptoken
ASSUMETARGETDEFS
USERID ggate, PASSWORD ggate
MAP sender.emptoken, TARGET receiver.emptoken,
COLMAP (USEDEFAULTS,
hostname = @token ('tk_host'),
osuser= @token ('tk_osuser'),
dbname= @token ('tk_dbname'),
tran_date = @DATENOW());

Now Test the data…

On Source

SQL> connect sender/sender

```
Connected.
SQL> select * from emptoken
  2  ;

    EMPID EMPNAME
--------- ----------
      100 TESTTOKEN

SQL> insert into emptoken values(101,'TESTTOKEN1');
1 row created.

SQL> COMMIT;
Commit complete.
```

On Target

```
SQL> select * from receiver.emptoken
  2  ;

    EMPID EMPNAME    HOSTNAME           OSUSER     DBNAME     TRAN_DATE
--------- ---------- ------------------ ---------- ---------- ---------
      100 TESTTOKEN  oracledb           oracle     SOURCE     15-APR-16
      101 TESTTOKEN1 oracledb           oracle     SOURCE     15-APR-16
```

Exercise 9:

Steps for performing converting Classic capture to an Integrated Capture (12c New Features)

For practice, use Exercise 7 – GoldenGate extract process **ext2** & Replicat **rep2** processes

To upgrade the classic capture to an integrated capture, we need to stop the extract process first and register it with the database

On Source

GGSCI (oracledb) 7> stop extract ext2

Sending STOP request to EXTRACT EXT2 ...
Request processed.

GGSCI (oracledb) 9> dblogin userid ggate password ggate

Successfully logged into database.

GGSCI (oracledb) 10> register extract ext2 database
Extract EXT2 successfully registered with database at SCN 1904137.

GGSCI (oracledb) 13> start ext2

Sending START request to MANAGER ...
EXTRACT EXT2 starting

Do some insert at source and ensure to replicat target side. This just makes sure that the GGSCI processes are still working

SQL> insert into test_obj(OWNER)values('&OWNER');
Enter value for owner: VASAN
old 1: insert into test_obj(OWNER)values('&OWNER')
new 1: insert into test_obj(OWNER)values('VASAN')

1 row created.

SQL> /
Enter value for owner: VASAN1
old 1: insert into test_obj(OWNER)values('&OWNER')
new 1: insert into test_obj(OWNER)values('VASAN1')

Check: Check at target these 2 records are successfully inserted.

Convert Target process before you convert the source process.

On Target

GGSCI (oracledb) 2> stop rep2

Sending STOP request to REPLICAT REP2 ...
Request processed.

GGSCI (oracledb) 3> alter replicat rep2,Integrated
REPLICAT (Integrated) altered.

GGSCI (oracledb) 6> start rep2

Sending START request to MANAGER ...
REPLICAT REP2 starting

GGSCI (oracledb) 9> info rep2

REPLICAT REP2 Last Started 2016-04-15 13:52 Status RUNNING
INTEGRATED
Checkpoint Lag 00:00:00 (updated 00:19:20 ago)
Process ID 17433
Log Read Checkpoint File
/u01/app/oracle/product/12.1.0/target/dirdat/handle/rt000002
2016-04-15 13:29:02.000533 RBA 2097

Do some insert at source and ensure to replicat target side

Check the running report on (Source & Target)

On Target

GGSCI (oracledb) 2> view report REP2

2016-04-15 13:52:05 INFO OGG-**02527 Integrated Replicat** does not populate a trace table.

2016-04-15 13:52:05 INFO OGG-02545 Parameter GROUPTRANSOPS is ignored by Integrated Replicat when parallelism is greater than 1.

2016-04-15 13:52:10 INFO OGG-02528 REPLICAT REP2 successfully **registered with database as inbound server OGG$REP2.**

2016-04-15 13:52:13 INFO OGG-02530 Integrated **replicat successfully attached to inbound server OGG$REP2.**

****
**
** Run Time Messages **
****
**

On Source

GGSCI (oracledb) 4> **info ext2**

EXTRACT EXT2 Last Started 2016-04-15 13:48 Status RUNNING

Checkpoint Lag 00:00:09 (updated 00:00:02 ago)
Process ID 17310
Log Read Checkpoint Oracle **Integrated Redo Logs**
2016-04-15 15:06:52
SCN 0.1931733 (1931733)

GGSCI (oracledb) 2> **view report ext2**

016-04-15 13:48:34 INFO OGG-02068 **Integrated capture successfully attached to logmining server OGG$CAP_EXT2 using OGGCapture API.**

2016-04-15 13:48:34 **INFO OGG-02086 Integrated Dictionary will be used.**

2016-04-15 13:48:39 INFO OGG-01226 Socket buffer size set to 27985 (flush size 27985).

Exercise 10:

Steps to create credential store (12c New Features)

Add the wallet and credential store information (password) for user ggate on both source and target systems

On Source

GGSCI (oracledb) 2> create wallet

Created wallet at location 'dirwlt'.

Opened wallet at location 'dirwlt'.

GGSCI (oracledb) 3> add credentialstore

Credential store created in ./dircrd/.

GGSCI (oracledb) 5> alter credentialstore add user ggate@source password gga te alias ggsource

Credential store in ./dircrd/ altered.

GGSCI (oracledb) 6> alter credentialstore add user ggate@target password gga te alias ggtarget

Credential store in ./dircrd/ altered.

GGSCI (oracledb) 7> info credentialstore

Reading from ./dircrd/:

Domain: OracleGoldenGate

Alias: ggsource
Userid: ggate@source

Alias: ggtarget
Userid: ggate@target

Note : The Wallet will be used for parameter files and also for DBLogin.

If you make a mistake entering wallet user, the command to remove a user is:
GGSCI> Alter CredentialStore Delete User username

Copy the source wallet single sign-on files to the target system. Execute the two copies

[oracle@oracledb dircrd]$ ls -ltr
total 4
-rw-r-----. 1 oracle oinstall 701 Apr 15 19:13 **cwallet.sso**
[oracle@oracledb dircrd]$ cp * /u01/app/oracle/product/12.1.0/target/dircrd/

[oracle@oracledb dirwlt]$ ls -ltr
total 4
-rw-r-----. 1 oracle oinstall 290 Apr 15 19:10 **cwallet.sso**
[oracle@oracledb dirwlt]$ cp * /u01/app/oracle/product/12.1.0/target/dirwlt

[oracle@oracledb dirwlt]$ **ls -lrt /u01/app/oracle/product/12.1.0/target/*/*.sso**
-rw-r-----. 1 oracle oinstall 701 Apr 15 19:16 /u01/app/oracle/product/12.1.0/target/dircrd/cwallet.sso
-rw-r-----. 1 oracle oinstall 290 Apr 15 19:17 /u01/app/oracle/product/12.1.0/target/dirwlt/cwallet.sso

Example: Parameter configuration:

<u>On Source</u>

extract ext2
useridalias ggsource
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/handle/rt
TABLE sender.test_obj;

On Target

replicat rep2
ASSUMETARGETDEFS
useridalias ggtarget
MAP sender.test_obj, TARGET receiver.test_obj;

Exercise 11:

Steps for performing- Handling GoldenGate Exceptions and Errors with REPERROR

Use **REPERROR** parameter in Replicat parameter file to manage the way that the replication process responds to or handles any errors encountered in any of the DML statements which it is trying to process.

In the example we will see how we are handling the ORA-00001: unique constraint violated error using an exception handler specified via the **REPERROR (-1, EXCEPTION)** clause of the Replicat parameter file.

Prerequisite:

Create a directory for trail file's location.

$ mkdir /u01/app/oracle/product/12.1.0/target/dirdat/dmlexc

Create table empl_execption with additional columns in Target (receiver.empl_exception) Table (optype varchar2 (20), dberrnum varchar2 (20), dberrmsg varchar2 (20))

On Source

GGSCI> add extract extexc, tranlog, Begin now

GGSCI> edit params extexc

extract extexc

useridalias ggsource
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/extexc/rt
table sender.empl;

GGSCI> add rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/extexc/rt, extract extexc

On Target

GGSCI> add replicat repexc, exttrail /u01/app/oracle/product/12.1.0/target/dirdat/extexc/rt, checkpointtable ggate.chkptab

GGSCI> edit params repexc

```
REPLICAT  repexc
ASSUMETARGETDEFS
useridalias ggtarget
REPERROR (-1, EXCEPTION)
MAP sender.empl, TARGET receiver.empl;
INSERTALLRECORDS
MAP sender.empl, TARGET receiver.empl_exception,
EXCEPTIONSONLY,
COLMAP (USEDEFAULTS,
optype = @GETENV ("lasterr", "optype"),
dberr = @GETENV ("lasterr", "dberrnum"),
dberrmsg = @GETENV ("lasterr", "dberrmsg"));
```

On Source & Target

GGSCI > start extract extexc
GGSCI> start replicat repexc

So now we go and insert some duplicate rows on the source table.

```
SQL> insert into empl
  2   values
  3   (&1,'&2');
Enter value for 1: 2005
Enter value for 2: Testing
```

```
old   3:  (&1,'&2')
new   3:  (2005,'Testing')

1 row created.

SQL> commit;

Commit complete.
```

Check replicat process has not abended and continues to do the processing.

```
GGSCI (oracledb) 1> info replicat repexc

REPLICAT  REPEXC     Last Started 2016-05-22 10:26   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:00 ago)
Log Read Checkpoint  File /u01/app/oracle/product/12.1.0/target/dirdat/extexc/rt
/rt000003
2016-05-22 10:26:57.214525  RBA 5678
```

But the table EMP_EXCEPTIONS has been populated with the information about the duplicate rows which has caused the ORA-00001 error.

```
SQL> select * from emp_exception;

    EMPNO      EMPNAME      OPTYPE           DBERR            DBERRMSG
---------- ---------- -------------------- -------------------- --------------------
     2005         TESTING          INSERT           1              OCI Error ORA-00001
```

On Source & Target

GGSCI> view ggsevt (OR) view report <group name>

Exercise 12:

Steps for performing - Increasing Performance by Splitting Replication Loads (Sections 11)

On Source

```
[oracle@oracledb sqlscripts]$ echo $ORACLE_SID
source
[oracle@oracledb sqlscripts]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sun Apr 17 04:25:55 2016
```

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing opt
ions

SQL> connect sender/sender
Connected.
SQL> @range_split.sql
DROP   TABLE range_split
       *
ERROR at line 1:
ORA-00942: table or view does not exist
Table created.

SQL> @populate_range_split.sql
Procedure created.

GGSCI (oracledb) 2> dblogin useridalias ggsource
Successfully logged into database.

GGSCI (oracledb) 3> add trandata sender.range_split

Logging of supplemental redo data enabled for table SENDER.RANGE_SPLIT.
TRANDATA for scheduling columns has been added on table 'SENDER.RANGE_SPLIT
'.
GGSCI (oracledb) 4> info trandata sender.r*

Logging of supplemental redo log data is enabled for table SENDER.RANGE_SPL
IT.

Columns supplementally logged for table SENDER.RANGE_SPLIT: ROW_ID.

GGSCI (oracledb) 1> edit params defsrc

DefsFile /u01/app/oracle/product/12.1.0/source/dirdef/rangesplit.def, Purge
UserIDAlias ggsource
Table SENDER.RANGE_SPLIT;

[oracle@oracledb source]$ ./defgen paramfile /u01/app/oracle/product/12.1.0 /source/dirprm/defsrc.prm

```
********************************************************************
**
**          Running with the following parameters                 **
********************************************************************
**
DefsFile /u01/app/oracle/product/12.1.0/source/dirdef/rangesplit.def, Purge
UserIDAlias ggsource
Table SENDER.RANGE_SPLIT;
Retrieving definition for SENDER.RANGE_SPLIT.

Definitions generated for 1 table in /u01/app/oracle/product/12.1.0/source/
dirdef/rangesplit.def.
```

[oracle@oracledb dirdef]$ cp rangesplit.def /u01/app/oracle/product/12.1.0/target/dirdef/

On Target

```
[oracle@oracledb sqlscripts]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Sun Apr 17 04:46:48 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
SQL> connect receiver/receiver
Connected.
SQL> @range_split.sql
DROP   TABLE range_split
       *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

SQL>
```

On Source

GGSCI (oracledb) 1> edit params erangea

```
Extract erangea
UserIDAlias ggsource
RmtHost 192.168.0.35, MgrPort 7813
RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/ea
Table SENDER.RANGE_SPLIT, Filter (@RANGE (1, 3));
```

GGSCI (oracledb) 2> edit params erangeb

```
Extract erangeb
UserIDAlias ggsource
RmtHost 192.168.0.35, MgrPort 7813
RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/eb
Table SENDER.RANGE_SPLIT, Filter (@RANGE (2, 3));
```

GGSCI (oracledb) 2> edit params erangeb

```
Extract erangec
UserIDAlias ggsource
RmtHost 192.168.0.35, MgrPort 7813
RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/ec
Table SENDER.RANGE_SPLIT, Filter (@RANGE (3, 3));
```

GGSCI (oracledb) 4> add extract erangea, tranlog, begin now
EXTRACT added.

GGSCI (oracledb) 10> add extract erangeb, tranlog, begin now
EXTRACT added.

GGSCI (oracledb) 11> add extract erangec, tranlog, begin now
EXTRACT added.

GGSCI (oracledb) 7> add rmttrail
/u01/app/oracle/product/12.1.0/target/dirdat/ea,extract erangea
RMTTRAIL added.

GGSCI (oracledb) 12> add rmttrail
/u01/app/oracle/product/12.1.0/target/dirdat/eb,extract erangeb
RMTTRAIL added.

GGSCI (oracledb) 13> add rmttrail
/u01/app/oracle/product/12.1.0/target/dirdat/ec,extract erangec

RMTTRAIL added.

GGSCI (oracledb) 17> info all

Program Status Group Lag at Chkpt Time Since Chkpt

MANAGER RUNNING
EXTRACT STOPPED EFUNCS 00:00:00 00:49:25
EXTRACT STOPPED ERANGEA 00:00:00 00:06:17
EXTRACT STOPPED ERANGEB 00:00:00 00:03:50
EXTRACT STOPPED ERANGEC 00:00:00 00:03:41
EXTRACT STOPPED ETTOKEN 00:00:00 40:15:40
EXTRACT ABENDED EXT1 00:00:10 644:50:09
EXTRACT ABENDED EXT2 00:00:09 31:46:59
EXTRACT ABENDED OCCEXT 00:00:00 203:48:30
EXTRACT STOPPED PFUNCA 00:00:00 00:49:31
EXTRACT STOPPED PFUNCS 00:00:00 00:49:17

On Target

GGSCI (oracledb) 2> edit params rrangea

Replicat rrangea
UserIDAlias ggtarget
SourceDefs /u01/app/oracle/product/12.1.0/target/dirdef/rangesplit.def
DiscardFile /u01/app/oracle/product/12.1.0/target/dirrpt/rrangea.dsc, Append
Map SENDER.RANGE_SPLIT, Target RECEIVER.RANGE_SPLIT;

GGSCI (oracledb) 3> edit params rrangeb
Replicat rrangeb
UserIDAlias ggtarget
SourceDefs /u01/app/oracle/product/12.1.0/target/dirdef/rangesplit.def
DiscardFile ./u01/app/oracle/product/12.1.0/target/dirrpt/rrangeb.dsc, Append
Map SENDER.RANGE_SPLIT, Target RECEIVER.RANGE_SPLIT;

GGSCI (oracledb) 4> edit params rrangec

Replicat rrangec
UserIDAlias ggtarget
SourceDefs /u01/app/oracle/product/12.1.0/target/dirdef/rangesplit.def
DiscardFile ./u01/app/oracle/product/12.1.0/target/dirrpt/rrangec.dsc, Append
Map SENDER.RANGE_SPLIT, Target RECEIVER.RANGE_SPLIT;

GGSCI (oracledb) 7> add replicat rrangea,exttrail /u01/app/oracle/product/12.1.0/target/dirdat/ea, checkpointtable ggate.chkptab
REPLICAT added.

GGSCI (oracledb) 9> add replicat rrangeb,exttrail /u01/app/oracle/product/12.1.0/target/dirdat/eb, checkpointtable ggate.chkptab
REPLICAT added.

GGSCI (oracledb) 10> add replicat rrangec,exttrail /u01/app/oracle/product/12.1.0/target/dirdat/ec, checkpointtable ggate.chkptab
REPLICAT added.

On Source

GGSCI (oracledb) 2> start ER*

Sending START request to MANAGER ...
EXTRACT ERANGEA starting

Sending START request to MANAGER ...
EXTRACT ERANGEB starting

Sending START request to MANAGER ...
EXTRACT ERANGEC starting

GGSCI (oracledb) 3> info all

Program Status Group Lag at Chkpt Time Since Chkpt

MANAGER RUNNING
EXTRACT STOPPED EFUNCS 00:00:00 01:44:29
EXTRACT RUNNING ERANGEA 00:00:00 01:01:21
EXTRACT RUNNING ERANGEB 00:00:00 00:58:53
EXTRACT RUNNING ERANGEC 00:00:00 00:58:44
EXTRACT STOPPED ETTOKEN 00:00:00 41:10:43
EXTRACT ABENDED EXT1 00:00:10 645:45:12
EXTRACT ABENDED EXT2 00:00:09 32:42:02
EXTRACT ABENDED OCCEXT 00:00:00 204:43:33
EXTRACT STOPPED PFUNCA 00:00:00 01:44:34
EXTRACT STOPPED PFUNCS 00:00:00 01:44:21

On Target

GGSCI (oracledb) 12> start rrange*

Sending START request to MANAGER ...
REPLICAT RRANGEA starting

Sending START request to MANAGER ...
REPLICAT RRANGEB starting

Sending START request to MANAGER ...
REPLICAT RRANGEC starting

GGSCI (oracledb) 29> info all

Program Status Group Lag at Chkpt Time Since Chkpt

MANAGER RUNNING
REPLICAT ABENDED OCCREP 00:00:00 204:48:08
REPLICAT ABENDED REP1 00:00:00 645:49:48
REPLICAT ABENDED REP2 00:00:00 32:46:28
REPLICAT STOPPED RFUNC 00:00:00 01:47:30
REPLICAT STOPPED RFUNCA 00:00:00 01:47:24
REPLICAT STOPPED RPTOKEN 00:00:00 41:15:27
REPLICAT RUNNING RRANGEA 00:00:00 00:00:05
REPLICAT RUNNING RRANGEB 00:00:00 00:00:04
REPLICAT RUNNING RRANGEC 00:00:00 00:00:02

On Source

Connect to source database with sender user schema

SQL> connect sender/sender
Connected.
SQL> exec populate_range_split(500000,1000);

GGSCI (oracledb) 2> stats extract ERANGEA

Sending STATS request to EXTRACT ERANGEA ...

Start of Statistics at 2016-04-17 06:15:08.

Output to /u01/app/oracle/product/12.1.0/target/dirdat/ea:

Extracting from SENDER.RANGE_SPLIT to SENDER.RANGE_SPLIT:

*** Total statistics since 2016-04-17 06:13:19 ***

Total inserts	36523.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	36523.00

GGSCI (oracledb) 3> stats extract ERANGEA

Sending STATS request to EXTRACT ERANGEA ...

Start of Statistics at 2016-04-17 06:15:29.

Output to /u01/app/oracle/product/12.1.0/target/dirdat/ea:

Extracting from SENDER.RANGE_SPLIT to SENDER.RANGE_SPLIT:

*** Total statistics since 2016-04-17 06:13:19 ***

Total inserts	43180.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	43180.00

GGSCI (oracledb) 4> stats extract ERANGEb

Sending STATS request to EXTRACT ERANGEB ...

Start of Statistics at 2016-04-17 06:15:40.

Output to /u01/app/oracle/product/12.1.0/target/dirdat/eb:

Extracting from SENDER.RANGE_SPLIT to SENDER.RANGE_SPLIT:

*** Total statistics since 2016-04-17 06:13:19 ***

Total inserts	46115.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	46115.00

<u>On Target</u>

Check stats of all replicat in target

GGSCI (oracledb) 40> stats replicat RRANGEA
GGSCI (oracledb) 41> stats replicat RRANGEB
GGSCI (oracledb) 42> stats replicat RRANGEC

Exercise 13:

Steps for performing - Increasing Performance by Coordinated Applies (Section 11)

On Source

GGSCI (oracledb) 6> stop ER*

SQL> connect sender/sender
Connected.

SQL> truncate table sender.range_Split;
Table truncated.

On Target

GGSCI (oracledb) 42> stop RR*

Sending STOP request to REPLICAT RRANGEA ...
Request processed.

Sending STOP request to REPLICAT RRANGEB ...
Request processed.

Sending STOP request to REPLICAT RRANGEC ...
Request processed.

SQL> connect receiver/receiver
Connected.
SQL> truncate table receiver.range_split;
Table truncated.

On Source

GGSCI (oracledb) 1> edit params ecord

Extract ecord
UserIDAlias ggsource
RmtHost 192.168.0.35, MgrPort 7813
RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/ed
Table SENDER.RANGE_SPLIT;

Note : Save and close the file. Note that there is nothing in the Extract that indicates anything "coordinated" about the apply. The only difference between this Extract and the previous three was removing the Table clause:

, Filter (@RANGE (n, 3))

and the fact that you needed three of them. Now we only need one Extract

GGSCI (oracledb) 2> add extract ecord, tranlog,begin now
EXTRACT added.

GGSCI (oracledb) 3> add rmttrail /u01/app/oracle/product/12.1.0/target/dir dat/ed, extract ecord
RMTTRAIL added.

GGSCI (oracledb) 4> start ER ec*

Sending START request to MANAGER ...
EXTRACT ECORD starting

<u>On Target</u>

GGSCI (oracledb) 2> edit params rcord
Replicat rcord
UserIDAlias ggtarget
SourceDefs /u01/app/oracle/product/12.1.0/target/dirdef/rangesplit.def
DiscardFile /u01/app/oracle/product/12.1.0/target/dirrpt/rrangea.dsc, Append
Map SENDER.RANGE_SPLIT, Target RECEIVER.RANGE_SPLIT ThreadRange (1-3, ROW_ID);

GGSCI (oracledb) 4> add replicat rcord,coordinated,exttrail /u01/app/oracle/prod uct/12.1.0/target/ed, MaxThreads 3, checkpointtable ggate.chkptab
REPLICAT (Coordinated) added.

GGSCI (oracledb) 5> Start ER rc*

Sending START request to MANAGER ...

REPLICAT RCORD starting

GGSCI (oracledb) 6> info rcord

REPLICAT RCORD Last Started 2016-04-17 06:56 Status RUNNING
COORDINATED Coordinator MAXTHREADS 3
Checkpoint Lag 00:00:00 (updated 00:00:00 ago)
Process ID 15171
Log Read Checkpoint File /u01/app/oracle/product/12.1.0/target/ed000000
First Record RBA 0

GGSCI (oracledb) 6> info rcord

REPLICAT RCORD Last Started 2016-04-17 06:56 Status RUNNING
COORDINATED Coordinator MAXTHREADS 3
Checkpoint Lag 00:00:00 (updated 00:00:00 ago)
Process ID 15171
Log Read Checkpoint File /u01/app/oracle/product/12.1.0/target/ed000000
First Record RBA 0

GGSCI (oracledb) 7> Info rcord, Detail

REPLICAT RCORD Last Started 2016-04-17 06:56 Status RUNNING
COORDINATED Coordinator MAXTHREADS 3
Checkpoint Lag 00:00:00 (updated 00:00:09 ago)
Process ID 15171
Log Read Checkpoint File /u01/app/oracle/product/12.1.0/target/ed000000
First Record RBA 0

Lowest Log BSN value: (requires database login)

Active Threads:
ID Group Name PID Status Lag at Chkpt Time Since Chkpt
1 RCORD001 15180 RUNNING 00:00:00 00:00:00
2 RCORD002 15181 RUNNING 00:00:00 00:00:00
3 RCORD003 15182 RUNNING 00:00:00 00:00:00

Current directory /u01/app/oracle/product/12.1.0/target

Report file /u01/app/oracle/product/12.1.0/target/dirrpt/RCORD.rpt
Parameter file /u01/app/oracle/product/12.1.0/target/dirprm/rcord.prm
Checkpoint file /u01/app/oracle/product/12.1.0/target/dirchk/RCORD.cpr

Checkpoint table ggate.chkptab
Process file /u01/app/oracle/product/12.1.0/target/dirpcs/RCORD.pcr
Error log /u01/app/oracle/product/12.1.0/target/ggserr.log

GGSCI (oracledb) 8> Info rcord002

REPLICAT RCORD002 Last Started 2016-04-17 06:56 Status RUNNING
COORDINATED Replicat Thread **Thread 2**
Checkpoint Lag 00:00:00 (updated 00:00:06 ago)
Process ID 15181
Log Read Checkpoint File /u01/app/oracle/product/12.1.0/target/ed000000
First Record RBA 0

GGSCI (oracledb) 9> Info rcord003

REPLICAT RCORD003 Last Started 2016-04-17 06:56 Status RUNNING
COORDINATED Replicat Thread **Thread 3**
Checkpoint Lag 00:00:00 (updated 00:00:04 ago)
Process ID 15182
Log Read Checkpoint File /u01/app/oracle/product/12.1.0/target/ed000000
First Record RBA 0

GGSCI (oracledb) 10> Stats rcord - Give cumulative records

GGSCI (oracledb) 11> Stats rcord002 - Splited with each thread sharing equally the record process…

GGSCI (oracledb) 12> Stats rcord003

Exercise 14:

Steps for performing – Bidirectional Replication (resolving loop deduction)

Site A

Extract – extA

GGSCI > add extract **extA,** tranlog, begin now

GGSCI > edit params **extA**

extract extA
UserIDAlias ggsource
rmthost 192.168.0.35, mgrport 7813
rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/at
tranlogoptions excludeuser ggate
table sender.empl;

GGSCI> add rmttrail /u01/app/oracle/product/12.1.0/target/dirdat/at, extract **extA**

Site B

Replicat – repA

GGSCI> add replicat **repA**, exttrail /u01/app/oracle/product/12.1.0/target/dirdat/at, checkpointtable **ggate.chkptab**

GGSCI > edit params **repA**
replicat repA
UserIDAlias ggtarget
assumetargetdefs
MAP sender.empl,TARGET receiver.empl;

Extract **– ExtB**

GGSCI> add extract **extB,** tranlog, begin now

GGSCI> edit params **extB**
extract **extB**
UserIDAlias ggtarget
rmthost 192.168.0.35, mgrport 7812
rmttrail /u01/app/oracle/product/12.1.0/source/dirdat/bt
tranlogoptions excludeuser ggate
table receiver.empl;

GGSCI> add rmttrail /u01/app/oracle/product/12.1.0/source/dirdat/bt, extract **extB**

Site A

GGSCI> edit params .GLOBALS
Checkpointtable ggate.chkptab1

GGSCI> dblogin userid ggate,password ggate
GGSCI> add checkpointtable ggate.chkptab1

Replicat – **repB**

GGSCI> add replicat **repB**, exttrail /u01/app/oracle/product/12.1.0/source/dirdat/bt, checkpointtable **ggate.chkptab1**

GGSCI> edit params **repB**

replicat **repB**
UserIDAlias ggsource
assumetargetdefs
MAP receiver.empl, TARGET sender.empl;

Site A

GGSCI> Start extA
GGSCI > Start repB (do after start replication (repA) at Site B)

Site B

GGSCI> start extB
GGSCI > start repA

Exercise 15:

Steps for performing - GoldenGate Replication using Macros, User Tokens, Password Encryption, Trail Encryption and SQLEXEC

1: Set Macros and User Tokens

Oracle GoldenGate Macros provide functionality for sharing parameters or other runtime configuration settings across multiple components and externalizing complex configuration settings to streamline parameter file contents.

The best practice is to create a file or series of files as a macro library and store them in a specific folder (for example, $OGG_HOME/dirmac).

Edit the Macro files and update as below

- #dbconnect
- #bpsettings
- #funcsmap

<a> The keyword that starts a macro body

<b> The keyword that finishes a macro

<c> Reset the statistics when a new report is generated

<d> Generates a report every day at one minute after midnight

<e> Close the current report file and create a new one daily at one minute after midnight

<f> The keyword that maps records between different source and target columns

<g> The function that is used to return information about the Oracle GoldenGate environment

<h> The function used to identify a user token

Create a **macrolib.mac file** with below content

```
MACRO #dbconnect
<a> UserID ggate, Password <encrypted_pswd> AES256, EncryptKey MyKey3
 <b>;

MACRO #bpsettings BEGIN
<c>
<d>
<e>
ReportCount Every 60 Seconds, Rate
END;

MACRO #funcsmap
PARAMS (#src_table, #target_table)
BEGIN
        MAP #src_table, TARGET #target_table,
        <f> (usedefaults,
                gg_commit_ts = <g> ( 'GGHEADER' , 'COMMITTIMESTAMP' ),
                lag_extract_ms = <h> ( 'TKN-EXTLAG-MSEC' ),
                lag_replicat_ms = @GETENV ( 'LAG' , 'MSEC' ),
                src_db_name = @TOKEN ( 'TKN-SRC-DBNAME' ),
                src_db_version = @TOKEN ( 'TKN-SRC-DBVERSION' ),
                src_txn_csn = @TOKEN ( 'TKN-TXN-CSN' )
            );
        END
```

Copy the **macrolib.mac (Macro library file) into both source and target location under** /u01/app/oracle/product/12.1.0/source/dirmac **&** /u01/app/oracle/product/12.1.0/source/dirmac

$ mkdir /u01/app/oracle/product/12.1.0/source/dirmac

$ mkdir /u01/app/oracle/product/12.1.0/target/dirmac

On Source

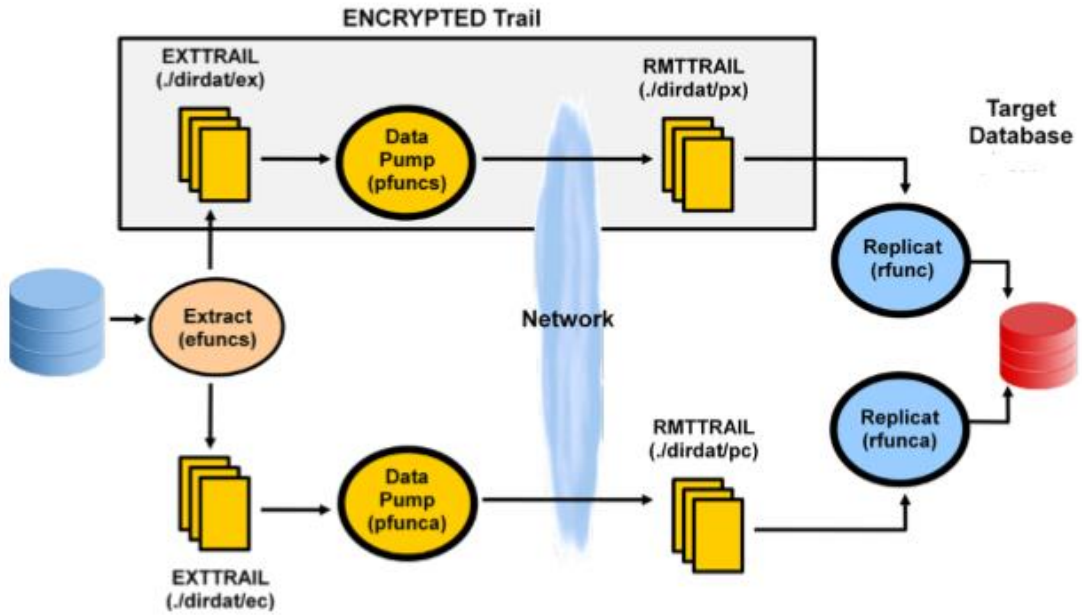
[oracle@oracledb dirmac]$ pwd
/u01/app/oracle/product/12.1.0/source/dirmac
[oracle@oracledb dirmac]$ ls -ltr *.mac

-rw-r--r--. 1 oracle oinstall 2930 Apr 17 02:34 macrolib.mac
[oracle@oracledb dirmac]$

On Target
[oracle@oracledb dirmac]$ pwd
/u01/app/oracle/product/12.1.0/target/dirmac
[oracle@oracledb dirmac]$ ls -ltr *.mac
-rw-r--r--. 1 oracle oinstall 2930 Apr 17 02:36 macrolib.mac

2: GoldenGate Replication using Macros, User Tokens, password encryption, trail encryption and SQLEXEC

Big Picture



Set Source & Target database tables for replication:

On Source
SQL> connect sender/sender
Connected.
SQL> @/home/oracle/Section13/sqlscripts/source_database.sql

On Target
SQL> connect receiver/receiver
Connected.
SQL> @/home/oracle/Section13/sqlscripts/target_database.sql

On Source

GGSCI (oracledb) 1> dblogin useridalias ggsource
Successfully logged into database.

GGSCI (oracledb) 2> add trandata sender.*

Generating Source Table Definitions:

GGSCI (oracledb) 2> edit params defgen

defsfile /u01/app/oracle/product/12.1.0/source/dirdef/strchg.defs, Purge
useridalias ggsource
table sender.wshop_encrypt;
table sender.wshop_funcs;
table sender.cust_zip;

[oracle@oracledb source]$ ./defgen paramfile /u01/app/oracle/product/12.1.0/source/dirprm/defgen.prm

[oracle@oracledb source]$ cp strchg.defs /u01/app/oracle/product/12.1.0/target/dirdef/

3: GoldenGate Encryption Using ENCKEYS

Oracle GoldenGate offers three types of encryption: Oracle GoldenGate Trail, Data Transmission, and password. In this practice, we will use AES256 FIPS-compliant encryption to encrypt the password for the database login.

Before data encryption can occur, encryption keys must be generated using keygen utility

To run keygen and create keys that will be used for password and data transmission encryption. In Source GoldenGate home directory, using keygen utility, create five keys, with one having 64 bits, one with 128 bits, and three having 256 bits. Then open the file with the text editor of your choice:

[oracle@oracledb source]$ ./keygen 64 1 > ENCKEYS

[oracle@oracledb source]$ ./keygen 128 1 >> ENCKEYS

[oracle@oracledb source]$ ./keygen 256 3 >> ENCKEYS

[oracle@oracledb source] vi ENCKEYS

(Name the keys **MyKey1** through **MyKey5** by adding the prefixes)
MyKey1 0x9015E54F4AB17A29

MyKey2 0x7527865E4705BE080CDBFE10A7A3CC4D

MyKey3
0xC3207D7D86FDE02B863EB332E7CD4E629BFB301B8D8F0528A15A270C5BB4BC75

MyKey4
0x1EF4C8498E859623A63F5A45C231DC40806AA76FFC769116B04A281A104FE54F

MyKey5
0x79C71416960D4C1BC64001589D95691F66D91D446B5E1D05BF3A2928C5E90D2A

Copy ENCKYS file into target GoldenGate Installation Location

[oracle@oracledb source]$ cp ENCKEYS /u01/app/oracle/product/12.1.0/target/

4: Password encryption using GoldenGate default Encryption

On Source

GGSCI (oracledb) 4> encrypt password ggate AES256 EncryptKey MyKey3

Encrypted password:
AADAAAAAAAAAAAJAMEHJTEOAXDFHGGSILHRGKAXIFJUGMCIEDECEUCXFAFWEWADFECLCEEUFA IVAJH QEIEKETECCOCJBAGJJGGAGYFRBVDLCHDIB

Algorithm used: AES256

Replace the hex decimal value in <encrypted_ password> entry in macro file.(macrolib.mac)

MACRO #dbconnect
BEGIN Userid gguser, Password
AADAAAAAAAAAAAJAMEHJTEOAXDFHGGSILHRGKAXIFJUGMCIEDECEUCXFAFWEWADFECLCEEUFA IVAJH QEIEKETECCOCJBAGJJGGAGYFRBVDLCHDIB AES256, EncryptKey MyKey3
END;

MACRO #bpsettings
BEGIN

Update the same value in Target Database macrolib.mac file
Now set the database replication …

On Source:

GGSCI (oracledb) 5> edit params efuncs

```
NoList
Include /u01/app/oracle/product/12.1.0/source/dirmac/macrolib.mac
List
Extract efuncs
EncryptTrail AES256 KeyName MyKey3
ExtTrail /u01/app/oracle/product/12.1.0/source/dirdat/ex

#dbconnect ()
#bpsettings ()
Table SENDER.WSHOP_ENCRYPT;

NoEncryptTrail
ExtTrail /u01/app/oracle/product/12.1.0/source/dirdat/ec
Table SENDER.WSHOP_ENCRYPT;
Table SENDER.CUST_ZIP;
Table SENDER.WSHOP_FUNCS, TOKENS (
   TKN-EXTLAG-MSEC = @GETENV ( 'LAG' , 'MSEC' ),
   TKN-SRC-DBNAME = @GETENV ( 'DBENVIRONMENT' , 'DBNAME' ),
   TKN-SRC-DBVERSION = @GETENV ( 'DBENVIRONMENT' , 'DBVERSION' ),
   TKN-TXN-CSN = @GETENV ( 'TRANSACTION' , 'CSN' )
);
```

GGSCI (oracledb) 6> dblogin useridalias ggsource
Successfully logged into database.

GGSCI (oracledb) 7> add extract efuncs,tranlog,begin now

GGSCI (oracledb) 8> Add ExtTrail /u01/app/oracle/product/12.1.0/source/dirdat/ex, Extract efuncs EXTTRAIL added.

GGSCI (oracledb) 9> Add ExtTrail /u01/app/oracle/product/12.1.0/source/dirdat/ec, Extract efuncs EXTTRAIL added.

Notice this Extract group will have two trails: one encrypted (ex) and one plain text (ec).

Configure Pump processes

GGSCI (oracledb) 10> edit params pfuncs
Extract **pfuncs**
RmtHost 192.168.0.35, MgrPort 7813, Compress
RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/px
Passthru
Table SENDER.*;

GGSCI (oracledb) 11> Add Extract pfuncs, ExtTrailsource /u01/app/oracle/product/12.1.0/source/dirdat/ex

GGSCI (oracledb) 12> Add RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/px, Extract pfuncs

Second Pump

GGSCI (oracledb) 8> edit params pfunca
Extract **pfunca**
RmtHost 192.168.0.35, MgrPort 7813, Compress
RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/pc
Passthru
Table SENDER.*;

GGSCI (oracledb) 9> Add Extract pfunca, ExtTrailsource /u01/app/oracle/product/12.1.0/source/dirdat/ec

GGSCI (oracledb) 10> add RmtTrail /u01/app/oracle/product/12.1.0/target/dirdat/pc, Extract pfunca

On Target

Setting Replicat Parameter

GGSCI (oracledb) 1> edit params rfunca

```
NoList
Include /u01/app/oracle/product/12.1.0/target/dirmac/macrolib.mac
List

Replicat rfunca
#dbconnect ()
SourceDefs /u01/app/oracle/product/12.1.0/target/dirdef/strchg.defs
DiscardFile /u01/app/oracle/product/12.1.0/target/dirrpt/rfunca.dsc, purge

#bpsettings ()

#funcsmap (SENDER.WSHOP_FUNCS, RECEIVER.WSHOP_FUNCS)

Map SENDER.CUST_ZIP, TARGET RECEIVER.CUST_CITY_STATE,
SQLEXEC (ID ZIPLKUP,
      QUERY ' SELECT zip_city, zip_state FROM receiver.zip_lookup WHERE zip = :vzip
',
      PARAMS (vzip = cust_zip)),
ColMap (usedefaults,
     cust_city = @GETVAL (ZIPLKUP.zip_city),
     cust_state = @GETVAL (ZIPLKUP.zip_state)
```

GGSCI (oracledb) 2> edit params rfunc

```
GGSCI (oracledb) 2> edit params rfunc
Replicat rfunc
UserIDAlias ggtarget
AssumeTargetDefs
DiscardFile /u01/app/oracle/product/12.1.0/target/dirrpt/rfunc.dsc, Purge
DecryptTrail AES256 KeyName MyKey3
Map SENDER.*, Target RECEIVER.*;

GGSCI (oracledb) 3> Add Replicat rfunc, ExtTrail
/u01/app/oracle/product/12.1.0/target/dirdat/px, checkpointtable ggate.chkptab
```

GGSCI (oracledb) 4> Add Replicat rfunca, ExtTrail /u01/app/oracle/product/12.1.0/target/dirdat/pc, checkpointtable ggate.chkptab

On Source

GGSCI (oracledb) 9> start ER ef*

GGSCI (oracledb) 9> start ER pf*

On Target

GGSCI (oracledb) 3> start ER rf*

5: Generating Transactions and Validating Results

On Source

Generate source database transactions by executing the following command

```
SQL> connect sender/sender
Connected.
SQL> @trans_generator.sql

SQL>  SELECT * FROM wshop_encrypt;

ROW_NUMBER ROW_TEXT
---------- ------------------------
         1 OXVFpcOPdFcSjiUkwLbaPFgna
         2 lGaTqmcdicEtBDlgEbBvSuFwc
         3 nImxJoIcfISBsHnEWcjXWdQCb
         4 LXbsXOuiFnvRmKHMLCvsUIyef
         5 pQiNfPTKYBLWWNNGxJtyLjomT
         6 PxYAGqNrbktJCPMxcIhVaRAFm
         7 SThTRJfglgVKYQdvQgQdTvmCf
         8 gJGmZBQsiZIXXIVZfNndbYYNl
         9 DkvroxArZDihweOvSvomRqLdR
        10 wXKasSNrhQpNtZzMFFFAoafjR

10 rows selected.

SQL>
```

On Target

```
SQL>  SELECT * FROM wshop_encrypt;

ROW_NUMBER ROW_TEXT
---------- ------------------------
         1 OXVFpcOPdFcSjiUkwLbaPFgna
         2 lGaTqmcdicEtBDlgEbBvSuFwc
         3 nImxJoIcfISBsHnEWcjXWdQCb
         4 LXbsXOuiFnvRmKHMLCvsUIyef
         5 pQiNfPTKYBLWWNNGxJtyLjomT
         6 PxYAGqNrbktJCPMxcIhVaRAFm
         7 SThTRJfglgVKYQdvQgQdTvmCf
         8 gJGmZBQsiZIXXIVZfNndbYYNl
         9 DkvroxArZDihweOvSvomRqLdR
        10 wXKasSNrhQpNtZzMFFFAoafjR

10 rows selected.

SQL>
```

Validating the Token, Macro, and SQLEXEC Information

<u>On Target</u>

Token:

```
SQL> SELECT src_db_name, src_db_version, src_txn_csn    FROM wshop_funcs;

SRC_DB_NAM SRC_DB_VERSION                                  SRC_TXN_CSN
---------- ------------------------------------------------ -----------
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099485
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099487
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099489
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099491
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099493
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099495
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099497
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099475
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099479
SOURCE     Oracle Database 12c Enterprise Edition Release 12.1.0.2     2099483

10 rows selected.

SQL>
```

```
SQL> SELECT lag_extract_ms, lag_replicat_ms          FROM wshop_funcs;

LAG_EXTRACT_MS LAG_REPLICAT_MS
-------------- ---------------
          3843          372548
          3843          372612
          3843          372612
          3843          372612
          3843          372612
          3843          372612
          3843          372612
          3843          372548
          3843          372548
          3843          372548

10 rows selected.

SQL>  SELECT gg_commit_ts FROM wshop_funcs;

GG_COMMIT_TS
-------------------------------------------------------------------------------
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM
17-APR-16 03.44.04.001847 AM

10 rows selected.
```

SQLEXEC

```
SQL> SELECT * FROM cust_city_state;

   CUST_ID CUST_CITY               CU   CUST_ZIP
---------- ------------------------ -- ----------
         1 Wheat Ridge              CO      80033
         2 New Orleans              LA      70117
         3 San Francisco            CA      94105
```

```
4 Denver          CO    80202
5 Jefferson       LA    70001

SQL>
```

Exercise 16:

Steps for performing - Encryption Using Wallets

In previous practices, we used ENCKEYS to manually maintain encryption keys. In this practice, you will use the Wallet to maintain encryption keys. Wallet keys are the preferred method of specifying trail encryption

On Source

```
GGSCI (oracledb) 1> open wallet
Opened wallet at location 'dirwlt'.

GGSCI (oracledb) 2> add Masterkey
Master key 'OGG_DEFAULT_MASTERKEY' added to wallet at location 'dirwlt'.

GGSCI (oracledb) 3> info masterkey
Masterkey Name:            OGG_DEFAULT_MASTERKEY
Creation Date:             Sun Apr 17 13:42:28 2016

Version:      Creation Date:              Status:
1             Sun Apr 17 13:42:28 2016    Current

GGSCI (oracledb) 4>
```

Copy from source to target:

```
GGSCI (oracledb) 4> sh cp
/u01/app/oracle/product/12.1.0/source/dircrd/cwallet.sso
/u01/app/oracle/product/12.1.0/target/dircrd/cwallet.sso


GGSCI (oracledb) 5>  sh cp
/u01/app/oracle/product/12.1.0/source/dirwlt/cwallet.sso
/u01/app/oracle/product/12.1.0/target/dirwlt/cwallet.sso
```

On Target

Validate:

GGSCI (oracledb) 6> Open Wallet

Opened wallet at location 'dirwlt'.

GGSCI (oracledb) 7> Info MasterKey
Masterkey Name: OGG_DEFAULT_MASTERKEY
Creation Date: Sun Apr 17 13:42:28 2016

Version: Creation Date: Status:
1 Sun Apr 17 13:42:28 2016 Current

On Source

GGSCI (oracledb) 57> edit params **EXTSEND**
Extract extsend
ExtTrail /u01/app/oracle/product/12.1.0/source/dirdat/ew
UserIDAlias ggsource
Table SENDER.*;

GGSCI (oracledb) 58> edit params **psend**
Extract psend
EncryptTrail AES256
RmtHost 192.168.0.35, MgrPort 7813, Compress
RmtTrail /u01/app/oracle/product/12.1.0/targe/dirdat/pe
Passthru
Table SENDER.*;

GGSCI (oracledb) 59> Add Extract extsend, TranLog, Begin Now

GGSCI (oracledb) 60> Add ExtTrail
/u01/app/oracle/product/12.1.0/source/dirdat/ew, Extract extsend

GGSCI (oracledb) 59> Add Extract psend, ExtTrailSource
/u01/app/oracle/product/12.1.0/source/dirdat/ew

GGSCI (oracledb) 59> Add RmtTrail
/u01/app/oracle/product/12.1.0/target/dirdat/pe, Extract psend

On Target

GGSCI (oracledb) 3> edit params rrece
replicat rrece
AssumeTargetDefs
DiscardFile /u01/app/oracle/product/12.1.0/target/dirrpt/reast.dsc, Purge
UserIDAlias ggtarget
Map SENDER.*, Target RECEIVER.*;

GGSCI (oracledb) 6> add replicat rrece,exttrail /u01/app/oracle/product/12.1.0/target/dirdat/pe, checkpointtable ggate.chkptab
REPLICAT added.

GGSCI (oracledb) 7> start rrece

Sending START request to MANAGER ...
REPLICAT RRECE starting

On Source:

Insert some records and verify the data at target site. Also view report

GGSCI (oracledb) 59> view report psend

****
** Oracle GoldenGate Capture for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:47:30
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

……INFO OGG-05519 Output trail file **encryption: AES256**