Creating PostgreSQL databases and running ETL (Kettle) Jobs to load data.

Below are the steps to create new databases. The steps are after installing and setting up the Postgresql database 9.3.x or later. And all the steps below are for Linux environment.

1. Copy the source/database/POSTGRESQL folder from GitHub to /home/gpadmin/ folder.
2. If the source/database/POSTGRESQL folder from GitHub is not copied to /home/gpadmin then replace /home/gpadmin in the below files to the folder where the POSTGRESQL folder is copied.  
   a) POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/Checkbook\_DB\_SetUp.sh  
   b) POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/Scripts.sql  
   c) POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/ScriptsForReferenceTables.sql

d) POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/Trends.sql

e) POSTGRESQL/Checkbook/KETTLE\_JOB/PreProcessing\_DataFiles/master\_preprocess.sh

f) POSTGRESQL/Checkbook/KETTLE\_JOB/Solr/getSolrCount.sh

1. POSTGRESQL/Checkbook\_ogent/CREATE\_NEW\_DATABASE/Checkbook\_EDC\_DB\_SetUp.sh
2. POSTGRESQL/Checkbook\_ogent/CREATE\_NEW\_DATABASE/Scripts.sql
3. POSTGRESQL/Checkbook\_ogent/CREATE\_NEW\_DATABASE/ScriptsForReferenceTables.sql

j) POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/PreProcessing\_DataFiles/master\_preprocess.sh

k) POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/Solr/getSolrCount.sh  
l) POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/DUMP\_AND\_RESTORE/get\_oge\_contracts\_and\_restore.sh

m) POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/DUMP\_AND\_RESTORE/get\_oge\_fms\_data\_dump\_and\_restore.sh

n) POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/DUMP\_AND\_RESTORE/verify\_errors.sh

o) POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/ NYCCheckbookETL\_DDL.sql

p) POSTGRESQL/Checkbook\_ogent/CREATE\_NEW\_DATABASE/ NYCCheckbookETL\_DDL.sql

1. Login to the PostgreSQL database and create new database1 using the below command.  
    create database database\_name1;
2. Modify the parameter DB\_NAME in home/gpadmin/POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/Checkbook\_DB\_SetUp.sh file to the database name (database\_name1) created in the above (3rd) step.
3. Run the below command from home/gpadmin/POSTGRESQL/Checkbook/CREATE\_NEW\_DATABASE/ folder to create tables, procedures, initial reference/static data and Trends data. After running the below command verify if the database is successfully created or not.  
   nohup sh Checkbook\_DB\_SetUp.sh &
4. Login to the PostgreSQL database and create new database2 using the below command.  
    create database database\_name2;
5. Modify the parameter DB\_NAME in home/gpadmin/POSTGRESQL/Checkbook\_ogent/CREATE\_NEW\_DATABASE/Checkbook\_EDC\_DB\_SetUp.sh file to the database name (database\_name2) created in the above (6th) step.
6. Run the below command from home/gpadmin/POSTGRESQL/Checkbook\_ogent/CREATE\_NEW\_DATABASE/ folder to create tables, procedures, initial reference/static data. After running the below command verify if the database is successfully created or not.  
   nohup sh Checkbook\_EDC\_DB\_SetUp.sh &

Below are the steps to process test data using the ETL (Pentaho Kettle) Job.

1. Download the pentaho data integration from the below link in the same server where the POSTGRESQL is installed. Kettle requires the Sun Java Runtime Environment (JRE) version 1.5 <http://sourceforge.net/projects/pentaho/files/Data%20Integration/4.2.0-stable/>
2. Extract the tar file into folder of your choice, say /usr/local/ folder. It will create a folder called data-integration. So now the path of the kettle software is /usr/local/data-integration
3. Copy the POSTGRESQL/.kettle folder to the system user’s home directory.
4. Modify the below parameters with the correct values in the user\_home\_directory/.kettle/kettle.properties file  
     
   NYC\_EMAIL\_SERVER=MAIL\_SERVER\_NAME\_HERE

NYC\_EMAIL\_PORT=25

NYC\_EMAIL\_FROM=FROM\_EMAIL\_ID\_HERE

NYC\_EMAIL\_TO=TO\_EMAIL\_ID\_HERE

NYC\_EMAIL\_CC=CC\_EMAIL\_ID\_HERE

NYC\_FMS\_GROUP\_EMAIL\_CC=CC\_EMAIL\_ID\_HERE

NYC\_OASIS\_GROUP\_EMAIL\_CC=CC\_EMAIL\_ID\_HERE

POSTGRES\_CHECKBOOK\_DB\_HOST=DB\_HOSTNAME\_HERE

POSTGRES\_CHECKBOOK\_DB\_USER=DB\_USER\_HERE

POSTGRES\_CHECKBOOK\_DB\_PASS=DB\_PASSWORD\_HERE

POSTGRES\_CHECKBOOK\_DB\_NAME=DB\_NAME\_HERE (database\_name1)

POSTGRES\_CHECKBOOK\_DB\_NAME\_OGE=DB\_NAME\_HERE (database\_name2)

POSTGRES\_CB\_FILES\_SOURCE\_DIR=/home/gpadmin/POSTGRESQL/Checkbook/SOURCE\_DATA/

POSTGRES\_CB\_FILES\_DEST\_DIR=/home/gpadmin/POSTGRESQL/Checkbook/DEST/

POSTGRES\_CB\_FILES\_GPDIST\_DIR=/home/gpadmin/POSTGRESQL/Checkbook/GPFDIST/datafiles/

POSTGRES\_CB\_FILES\_BACKUP\_DIR=/home/gpadmin/POSTGRESQL/Checkbook/BACKUP/

POSTGRES\_CB\_FILES\_OGE\_SOURCE\_DIR=/home/gpadmin/POSTGRESQL/Checkbook\_ogent/SOURCE\_DATA/

POSTGRES\_CB\_FILES\_OGE\_DEST\_DIR=/home/gpadmin/POSTGRESQL/Checkbook\_ogent/DEST/

POSTGRES\_CB\_FILES\_OGE\_GPDIST\_DIR=/home/gpadmin/POSTGRESQL/Checkbook\_ogent/GPFDIST/datafiles/

POSTGRES\_CB\_FILES\_OGE\_BACKUP\_DIR=/home/gpadmin/POSTGRESQL/Checkbook\_ogent/BACKUP/

1. Modify the below solr properties

NYC\_SOLR\_FULL\_INDEXING\_OS=http://hostname:port/solrCoreName/dataimport?command=full-import&clean=true&jobID=0

NYC\_SOLR\_CHECK\_FULL\_INDEX\_STATUS\_OS=http://hostname:port/solrCoreName/dataimport/

NYC\_SOLR\_RECORDS\_COUNT\_OS=http://hostname:port/solrCoreName/select/?q=\*%3A\*&version=2.2&start=0&rows=10&indent=on

NYC\_SOLR\_DELETE\_RECORDS\_OGE\_OS=http://hostname:port/solrCoreName/update?stream.body=%3Cdelete%3E%3Cquery%3Eagency\_type:oge%3C/query%3E%3C/delete%3E&commit=true

NYC\_SOLR\_FULL\_INDEXING\_OGE\_OS=http://hostname:port/solrCoreName/dataimport?command=full-import&entity=contracts\_oge&entity=spending\_oge&clean=false&jobID=0

NYC\_SOLR\_CHECK\_INDEX\_STATUS\_OGE\_OS=http:hostname:port/solrCoreName/dataimport/

1. Modify POSTGRESQL/Checkbook/KETTLE\_JOB/Solr/getSolrCount.sh and POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/Solr/getSolrCount.sh files with the correct hostname, port and solrCoreName values.
2. Modify POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/DUMP\_AND\_RESTORE/get\_oge\_contracts\_and\_restore.sh and POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/DUMP\_AND\_RESTORE/get\_oge\_fms\_data\_dump\_and\_restore.sh files with the correct database name and connection parameters.
3. The source files that needs to be processed by ETL jobs should be in the below folders. And the filenames should match with actual\_pattern column of etl.ref\_file\_name\_pattern table  
    home/gpadmin/POSTGRESQL/Checkbook/SOURCE\_DATA/ 🡪 To process test files in database\_name1  
   home/gpadmin/POSTGRESQL/Checkbook\_ogent/SOURCE\_DATA/ 🡪 To process test files in database\_name2
4. The layout of the files should be based on the file layout documents of POSTGRESQL/Checkbook/Checkbook\_File\_Layouts/Checkbook\_File\_Layouts \_20120717 folder.
5. Below is the command to be run for processing the test files that are in home/gpadmin/POSTGRESQL/Checkbook/SOURCE\_DATA/ directory to database\_name1.   
   /usr/bin/nohup /bin/sh /home/gpadmin/POSTGRESQL/Checkbook/KETTLE\_JOB/process\_data\_using\_etl\_for\_database\_name1.sh &
6. Below is the command to be run for processing the test files that are in home/gpadmin/POSTGRESQL/Checkbook\_ogent/SOURCE\_DATA/ directory to database\_name2.

/usr/bin/nohup /bin/sh /home/gpadmin/POSTGRESQL/Checkbook\_ogent/KETTLE\_JOB/process\_data\_using\_etl\_for\_database\_name2.sh &