**Ora2pg migration [Windows]**

* After the installation of Ora2Pg, we have to make changes (i.e., configure) in the ora2pg.conf file or ora2pg\_dist.conf file accordingly to the data to be migrated.

**Steps for configurating the ora2pg Configuration File**

* **INPUT SECTION**

1.Set the Oracle home directory

ORACLE\_HOME C:\app\sn240\product\11.2.0\dbhome\_1

2.Set Oracle database connection (datasource, user, password)

ORACLE\_DSN dbi:Oracle:host=localhost;sid=orcl;port=1521

ORACLE\_USER data

ORACLE\_PWD data

[Ex: here I am doing migration from ‘localhost’, my sid is ‘orcl’ and the schema I have created in oracle is ‘data’ along with password ‘data’. I have inserted the tables in this schema which I want to migrate]

* **SCHEMA SECTION**

1.Set it to 1 to Export Oracle schema to PostgreSQL schema

EXPORT\_SCHEMA            1

2.Oracle schema/owner to use. if we want to export all the schema then comment the option SCHEMA or else specify the schema name which is required for export

SCHEMA  <SCHEMA\_NAME>

[Ex: here I am exporting schema data i.e., SCHEMA data]

3.Enable/disable the CREATE SCHEMA SQL order at starting of the output file.

CREATE\_SCHEMA 1

[Ex: here I am exporting only 1 schema i.e., data]

* **EXPORT SECTION**

Type of export. Values can be the following keyword. TYPE Variable specifies which object type need to be exported to .sql file. At time we can export only one type of object.

PACKAGE            Export packages

INSERT                  Export data from table as INSERT statement

COPY                     Export data from table as COPY statement

VIEW                     Export views

GRANT                 Export grants

SEQUENCE          Export sequences

TRIGGER              Export triggers

FUNCTION          Export functions

PROCEDURE       Export procedures

TABLESPACE        Export tablespace (PostgreSQL >= 8 only)

TYPE                      Export user defined Oracle types

PARTITION          Export range or list partition (PostgreSQL >= v8.4)

FDW                      Export table as foreign data wrapper tables

MVIEW                 Export materialized view as snapshot refresh view

QUERY                  Convert Oracle SQL queries from a file.

KETTLE                  Generate XML ktr template files to be used by Kettle.

DBLINK                Generate oracle foreign data wrapper server to use as dblink.

SYNONYM            Export Oracle's synonyms as views on other schema's objects.

DIRECTORY         Export Oracle's directories as external\_file extension objects.

LOAD                     Dispatch a list of queries over multiple PostgreSQl connections.

TEST                       perform a diff between Oracle and PostgreSQL database.

TEST\_VIEW        perform a count on both side of rows returned by views

[EX. TYPE TABLE INSERT -- For exporting table and its data as insert statement

TYPE FUNCTION -- For exporting Function

TYPE VIEW -- For Exporting View

TYPE TRIGGER-- For Exporting Trigger

TYPE  COPY-- PROCEDURE ]

* **OUTPUT SECTION**

[This is optional, if you want to export directly then we can give the set the postgres]

1.Define the following directive to send export directly to a PostgreSQL database. This will disable file output.

PG\_DSN dbi:Pg:dbname=postgres;host=localhost;port=5432

PG\_USER postgres

PG\_PWD postgres

[Ex: here iam migrating in localhost with the ‘postgres’ user and password as ‘postgres’]

2.Or we can generate the output file in the same folder of the ora2pg.conf file in the .sql format

OUTPUT output.sql

[Ex: Here we can give any name, if we export table and insert at-a-time then there will be two files i.e., ‘table-output.sql’ and ‘insert\_output.sql’]

* **Execute the ora2pg command**

We have to direct into the ora2pg folder in the program files in order to execute this command

ora2pg -p -c ora2pg.conf [after saving the ora2pg.conf file with the required, this command should be executed]

-p -- Enable PLSQL to PLPGSQL code conversion.

-c -- Configuration file

Once this command executes successfully, required object will be exported to .sql file. This file can be used to load data in the postgresql database.

* **Connecting to postgress**

1.Firstly we need to direct or change the directory to \*/bin folder of postgres.

2.We should move the output.sql files also into this directory before executing the command.

3.Alter statement under create statement of table in the output file should be commented. Since it throws an ‘schema already exists’ in the postgres.

4.Connect to postgress using the following command

psql -h localhost -p 5432 -U postgres -W postgres

5. Create a database where we want to load the data, or it will load or create schema under default database i.e., postgres.

CREATE DATABASE <databasename>;

6. Use the newly created database

\c <databasename>;

7. execute the .sql file

\i output.sql;

8. Once the import is completed, verification can be done by running the below command which will list all the restored database along with schema.

\d

* **Can now open the postgresql i.e., pgAdmin and see the imported data.**