**DATA MIGRATION RELATED TECHNIQUES, SCRIPTS**

1. **Migrate from SQL server to MYSQL steps:**

a) First check, analyze sql database (ex: work database) and db objects like tables, views etc that will be migrated to mysql db

with the same name as work database along with db objects

b) Now we need to create DSN to connect to the sql server from Mysql

c) For this, goto Control Panel-->Administrative tools-->ODBC Data sources(64 bit) and Run it as Administrator

d) click on system DSN --> Add--> select Sql server--> Finish click on it

e) Then on this screen, give name of the ODBC source ex: work\_sql

f) For server name: give your sql server name ( you can get it using command select @@servername in sql query window)

g) Copy and paste this server name and click on Next

h) For next screen, set authentication like Windows NT authentication or SQL server authentication as desired, generally we

give Windows NT authentication as its more secure and use certificate based security mechanism, so keep it

i) On next screen : change the default database to: select work database here from dropdown which we want to migrate, click Next

j) Click Finish, test datasource, and click OK...to get your system DSN created

k) Now go to your Mysql workbench and in that select from Database menu--> Migration Wizard--> Start Migration

l) On next screen, select source database system as SQL server, Connection method as ODBC Data source and keep DSN as work\_sql

that we have created just now

m) Username as sa and Test connection , then click on Next

n) From the target RDBMS connection parameter, select Local Instance MySQL80 and click on Test connect and verify it ok, click Next

o) Now select the database that you want to migrate ( that is work db) and click on Next

p) It will migrate all source tables, views etc, click on Next.. Next...Next ...Finish

q) Goto Mysql database and check this work database and its tables, views objects and underline data etc

or fire select queries to check data seen from desired tables to confirm the migration

r) So using this method we can migrate all db objects like tables, views etc at once

s) Another way of migration is by using SSRS package to migrate any specific table only, not all at once

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

**2) MySQL to SQL Server Database Migration**

a) We need two software tools here, first is MySQL connector (download and install it from google)

using ODBC driver for MySQL (Connector/ODBC) link to your machine

b) free download SSMA from google and install in your machine and open this SSMA tool

c) Now go to File and Create new project, give it name as ex: SqlMigration11 and select desired SQL server from dropdown like

SQL server 2012 or 2014 which is suitable or required as necessary and click on OK

d) first, we need to connect to MySQL server, it will display all database, now we need to connect SQL server and enter the name

of the database that you want to create (ex: test), keep windows authentication and click on connect, press yes,

it will create new dB, click on continue and you can see this new db. test created in mssql there

e) now from MySQL, select database which you want to migrate and right click and select option Convert schema and click ok

f) Now from Mssql, select test db., go to Tools option and click on synchronize with database option

g) it will display all the details about your db. test and tables, press OK

h) Now uncheck this test db. from Mssql and select the source db. from MySQL and go to Tool--> Migrate data and click on it

i) keep the standard mode, driver etc. as selected by default, check server port, username, give pwd and click on connect, connect

j) and the migration is completed, now go to Mssql db. test, and see tables and data there and verify data to confirm the migration

--------------------------------------------------------------------------------

**3) Migrate SQL Server to Oracle using Oracle SQL Developer- sample steps**

a) First install both MSSQL and Oracle database server in your local system

b) We have MSSQL\_MIGRATION as sample database having some tables which we want to migrate to oracle

c) Now inside ORACLE SQL developer tool first connect to this MSSQL server db. by giving proper conn name, username/pwd.

hostname may be localhost or as desired, give desired port, and select MSSQL\_MIGRATION db. from Retrieve database option and

test this connection

d) Also create new db. schema connection say MIGRATION\_REPO inside SQL developer which would be our target oracle db. schema

where we want to migrate data, so now we have source and target schema connections are ready

e) Now select source schema that is MSSQL\_MIGRATION right click on it, select option--> Migrate to Oracle and click Next

f) On next screen, select MIGRATION\_REPO db. from connection dropdown and click on Next

g) On next screen, give name to this project as SAMPLE\_MIGRATION and also give your output directory path as you want to save output

directory script files

h) On the next screen, choose third party connection as SQL server and click next

i) Then select MSSQL\_MIGRATION from selected database and click Next...Next...Next

j) On the Target database screen, select MIGRATION\_REPO database from the dropdown, click next and finish

k) now this will load all db. tables and other objects into target db. schema

l) check into MIGRATION\_REPO schema we can only migration tabled there that are migrated.

m) You can create another new schema connection inside Oracle SQL developer tool same as MSSQL\_MIGRATION

with its login credentials and then connect and can have all MSSQL db. tables and other objects data in this schema

-------------------------------------------------------------------------------------------

**4) Oracle to MSSQL db. migration**

a) First download and install MSSSQL server migration assistant for oracle say ex: v5.2

b) After downloading this tool, inside its folder two exe files are there related to Oracle extension pack etc., install both

c) choose db. instance name as desired from the drop down, give server name and conn details with windows authentication and click

next, give pwd and click Next, Next ...OK

d) Now open SSMA tool, click on new project, give name it as SqlMigration1 and select SQL server desired from Migration to option,

give location as necessary and click on OK

e) Now connect to Oracle, specify parameter for new conn to source, provide as Oracle client provider, mode-std mode,

give server name, and other login info and click on connect, check oracle conn established with tables and other details

f) Now connect to SQL server, give desired conn details as necessary, click on connect...connect

g) From oracle source db.(global), right click on it and select Convert schema option

h) Then from SQL server target db., right click on it, and select synchronize with database option, click on ok

i) Now from oracle source db., right click, select Migrate data option and give conn details and connect

j) check data migration report screen with all details, and close it

k) refresh mssql db., and then check tables data and other objects data, you can see rows loaded there to complete migration

--------------------------------------------------------------------------------

**5) MSSQL to PostgreSQL migration and PostgreSQL to MSSQL migration**

a) From PostgreSQL db., click on + symbol--> go to Data source--> Microsoft SQL server,

give the conn details like in instance give desired SQL server db. name (Adam), host as localhost or as desired,

give port, username (sa) and pwd, source db. name from SQL server as per reqd, click on Test connection and click ok

b) Now you can find this jdbc SQL server db. (Adam) inside PostgreSQL db. console window having desired tables etc.

c) Now stretch Adam db. and inside contents likes tables etc. from both the sides (mssql top window and PostgreSQL bottom window)

d) And then from mssql source Adam db., select specific table say empl table, drag and drop it inside public schema of

Adam db. of PostgreSQL target db., and it will simply get loaded or migrated there inside target db. schema.

e) Same way drag other source db. tables and drop it into public schema of Adam db. and continue this process for all tables and

other objects that you want to migrate, here table along with data get migrated into target db. schema

f) this is simplest way of migration from mssql to PostgreSQL, then check/verify data to confirm it

g) Same way you can migrate tables and other objects from PostgreSQL db. to mssql db. schema by drag and drop from

PostgreSQL window to mssql window inside desired db. schema to migrate db. objects with data from PostgreSQL to mssql

so here we migrate data from both the ends in one tool using this simple method

-----------------------------------------------------------------------------------

**6) MSSQL to csv file automatically download using batch file**

Suppose you have selected query from mssql like below

select \* from [CustomerInvoices] where convert(date,Invoicedate) = convert(date,getdate())

We can export data of this query or it may be whole table data to csv automatically

using some batch file (Export.bat) SQL command, so that csv file will get automatically downloaded at desired folder path,

the batch file cmd would be as below

SQLCMD -S ChangeToServerName -d ChangeToDatabaseName -E -Q "select \* from [CustomerInvoices] where convert(date,invoicedate)=convert(date,getdate())" -s "," -o "D:\Result\Excel Export\DailySales.csv"

Here for date column used type date to avoid time comparison so that it will show my daily sales report as of today

using SQL command, it takes some parameters as below

start with SQL command, then your server name (-S), then db. name (-d), then trusted conn (-E) or here you can

give username/pwd, then query (-Q), then my separator ( -s) used as comma, and then path to my output file (-o)

Once we run the batch file, it will create the csv file at the desired path,

you can create a Task scheduler that will run the batch file based on your customer requests, you can

change the frequency to run it once or twice a day

**7) Extracting csv file data from SQL Server using Python Script:**

import pyodbc

import pandas as pd

import os

from datetime import datetime

from plyer import notification

# create SQL connection

connection = pyodbc.connect(driver = '{ODBC Driver 17 for SQL Server}',

host = 'DESKTOP-NAKP5E5',

database = "Test",

trusted\_connection = 'yes')

# SQL Command to read the data

sqlQuery = "select \* from dbo.SalesOrder where Region = 'South'"

# Getting the data from sql into pandas dataframe

df = pd.read\_sql(sql = sqlQuery, con = connection)

# Export the data on the Desktop

df.to\_csv(os.environ["userprofile"] + "\\Desktop\\PythonScript\\" + "SQL\_OrderData\_" +

datetime.now().strftime("%d-%b-%Y %H%M%S")

+ ".csv", index = False)

# Display Notifiction to User

notification.notify(title="Report Status!!!",

message=f"Sales data has been successfully saved into Excel.\

\nTotal Rows: {df.shape[0]}\nTotal Columns: {df.shape[1]}",

timeout = 10)

-------------------------------------------------------------------------------------------------------------------

**8) PostgreSQL to CSV file extract using pipe and comma delimiter**

Suppose you have PostgreSQL query like say select \* from Emps

and you want to export data of it to csv using simple query with some delimiter

then use below query for it from the query window and run it to get csv output file at desired path

COPY Emps TO 'F:\Examples 2019\psql\EmpOP.csv' DELIMITER '|' CSV HEADER

So here the csv file will get exported to this specified path with

pipe delimiter or you can use any other delimiter as reqd in it with header included

and so, the file will get exported to this path will all data

------------------------------------------------------------------------------------

**9) Fetch Data from MySQL and Save to CSV in Python**

import pandas as pd

import mysql.connector

db = mysql.connector.connect(user = 'root', database = 'research\_ms')

cursor = db.cursor()

query = "select text , user\_name from tweets\_data"

cursor.execute(query)

myallData = cursor.fetchall()

all\_user\_name = []

all\_tweets = []

for text,user\_name in myallData:

all\_user\_name.append(user\_name)

all\_tweets.append(text)

#we need to store this data to CSV

dic = {'user\_name' : all\_user\_name , 'tweets':all\_tweets}

df = pd.DataFrame (dic)

df\_csv = df.to\_csv('E:/Nasir\_Soft.csv')

----------------------------------------------------------------------------------

**10) Example to Export CSV Delimited by Pipe '|' and Without Quoted Fields from oracle**

# The following Python program will export CSV delimited with the pipe '|' and without quoting the fields.

import csv

import cx\_Oracle

con = cx\_Oracle.connect('hr/hrpsw@localhost/orcl')

cursor = con.cursor()

csv\_file = open("employees.csv", "w")

writer = csv.writer(csv\_file, delimiter='|', lineterminator="\n", quoting=csv.QUOTE\_NONE)

r = cursor.execute("SELECT \* FROM employees")

for row in cursor:

writer.writerow(row)

cursor.close()

con.close()

csv\_file.close()

-----------------------------------------------------------------------

**11) Export data from SQL Server to XML file**

Suppose we have some specific account table in mssql db and

have below query as:

sql: select \* from account

Now to get data in xml format, change this query as below:

select \* from account

for xml PATH('DATA') , root('UBLDATA');

you will see the result in xml format in well desired format

Same way you can extract data from other db query to xml file using above method

---------------------------------------------------------------------

**12) Update XML data using python**

Suppose we have simple xml file (Customer4.xml) as below having two records, the root tag name is customers,

under that child tag is customer, under it there is Place and Amount as two records

<Customers>

<customer>

<Place>INDIA</Place>

<Amount>4500</Amount>

</customer>

<customer>

<Place>USA</Place>

<Amount>9000</Amount>

</customer>

</Customers>

Now open a Python editor to write a python program file to update this xml file as below

import xml.etree.ElementTree as obj

def updateXML(filename):

tree=obj.ElementTree(file=filename)

root =tree.getroot()

for amounts in root.iter("Amount"):

amounts.text="5000"

tree=obj.ElementTree(root)

with open(filename,"wb") as fileupdate:

tree.write(fileupdate)

if \_\_name\_\_="\_\_main\_\_":

updateXML("Customer4.xml")

--------------------------------------------------------------------

**13) Import JSON data into SQL Server with 3 steps**

step a) import file using OPENROWSET

SELECT \* FROM OPENROWSET(BULK '', SINGLE\_CLOB) as import

Suppose your json file (named as cities.JSON) is stored in Downloads folder of your pc,

then check this file path, name and its contents, then give above query with proper path,name

to see its data as below

SELECT \*

FROM OPENROWSET (BULK 'C:\Users\DLCT\Downloads\cities.JSON', SINGLE\_CLOB) as import

Now this query is correct with proper path/name, so you can see its data

that you want to convert/load into sql server db table

step b) Convert JSON output from the variable into sql server table using below script, run in query window,

Here give proper file path, name etc

Declare @JSON varchar(max)

SELECT @JSON=BulkColumn

FROM OPENROWSET (BULK 'C:\Users\DLCT\Downloads\cities.JSON', SINGLE\_CLOB) as import

SELECT \*

FROM OPENJSON (@JSON)

step c) Import JSON file data into the sql server table

Declare @JSON varchar(max)

SELECT @JSON=BulkColumn

FROM OPENROWSET (BULK 'C:\Users\DLCT\Downloads\cities.JSON', SINGLE\_CLOB) as import

SELECT \* INTO cities

FROM OPENJSON (@JSON)

WITH

(

[rank] int,

[Name] varchar(50),

[Country] varchar(50),

[Population] int,

[Prev] int,

[Growth] varchar(50)

)

---- Here see below json input file for some of its contents, so that you can get idea

---- how its fields, data types used in above query

{

"rank": 1,

"Name": "Tokyo",

"Country": "Japan",

"Population": 34568972

"Prev": 373924156

"Growth": "-0.00145"

},

{

"rank": 2,

"Name": "Delhi",

"Country": "India",

"Population": 34568972

"Prev": 373924156

"Growth": "-0.00145"

},....

-------------------------------------------------------------------------------------

**14) JSON to Postgresql database**

1) Download and install JsontoPostgresql tool and open it

2) click on Wizard for Impoorting option and on the next screen

3) Open or select json file from dropdown box as source file, it will display records/data below

4) on next screen, select target schema, table, load type (Appen or Replace or update) as Replace and click next

5) Preview fields, and other details or select as you want, and click on import

6) Data will get imported into Postgresql database and then verify it

-------------------------------------------------------------------------------

**15) Import your JSON file to MYSQL DATABASE using python**

There are two ways to do this using normal technique or using panda, let us do this using normal way

First method using normal way:

1) Free download/install online JSON viewer or Pycharm tool to open our JSON file stored in folder

2) Suppose the file is opened and it has columns like person, company, year etc with data as per sample file format shown earlier

3) Install and open XAMMPP control panel tool, and open it and create new database with name it as JSON

4) Then create table with same column name and data types etc simillar to json file data

for ex: CREATE TABLE json\_file(person varchar(50), year varchar(50), company varchar(50))

and execute/run this command to create this table and select data from it to verify it with no data

5) Now open online json viewer, and create python file for data laod with some name like single\_data\_file.py and write below code in it

6)

import json

import pymysql ( our mysql db)

json\_data = open( "project3.josn"). read()

json\_obj = json.loads(json\_data)

con=pymysql.connect(host="localhost", user="root", password="", db="json")

cursor=con.cursor()

for item in json\_obj:

person= item.get("person")

year= item.get("person")

company= item.get("company")

cursor.execute("insert into json\_file(person,year,company) values (%s, %s, %s)", (person, year, company))

con.commit()

con.close()

------------------

and then save and run this .py file and then check the data gets loaded into desired table and verify data in mysql to cover this

---------------------------------------------

Second methond to do this using panda

1) create new python file with name like usingpandas.py openjsonviewer tool and write below code in it

2) install pandas if not there using cmd like pip install pandas (or using cmd) pip install sqlalchemy

and also install sqlalchemy, then from sqlalchemy we need to install pandas and create/establish connection,

then read json file and then load/replace data into our mysql db, you can either Append or replace data,

and if it has automatically created index column, then keep index as false and then run the file, see code below

3)

import pandas as pd

from sqlalchemy import create\_engine

engine = create\_engine('mysql+pymysql://root:password@localhost/json')

df=pd.read\_json("project3.json")

df.tosql("json\_file",con=engine,if\_exists='replace',index=False)

4) after running this usingpandas.py data will get imported into mysql , verify data

-----------------------------------------------------------------------------------------------------------

**16) How to send JSON object as a parameter to stored procedure**

CREATE PROCEDURE sp\_json\_type ( IN data JSON)

BEGIN

/\* declare variables \*/

DECLARE username VARCHAR(25) DEFAULT NULL;

DECLARE email VARCHAR(25) DEFAULT NULL;

/\* initialize data for variables\*/

SET username = json\_unquote(json\_extract(data,'&.name'));

SET email = json\_unquote(json\_extract(data,'&.email'));

/\*create user table and then insert data into it \*/

INSERT INTO user( name, email) values (username, email\_id);

END:

Compile this procedure and then call it

CALL sp\_json\_type('{"name":"mike","email":"mike@gmail.com"}');

and run this procedure and you can see 1 row affected or got loaded into user tbl

so verify it using command select \* from user to verify data

-------------------------------------------

**17 ) Import XSLX/CSV File To PostgreSQL database**

a) We are using pgAdmin4 command line editor for doing this load of xls data into postgresql

b) Our xlsx file having around 59k rows of data that we need to import, verify data

c) steps for doing it like first convert xlsx file to csv, then create table with same data structure, copy data and verify/select

ALSO GIVE All file permission (read/write) to this file to Everyone from its security tab properties so that we can execute it later

d) First open our xlsx file and save it as comma delimited csv file (Sample data set.csv) with some specific name in some folder

e) Open the pgAdmin4 tool which is mainly used to connect to postgresql db server, once its connected using login pwd etc

f) then go to the schema/db, public schema, right click and click on create script option where we srite script for doing this as below

f)

drop table public.department;

select \* from public.department; --- to see table is dropped

create table public.department(

summary\_date date,

legacy\_company char(50),

region\_name char(50),

management\_area char(50)

)

this table structure is simillar to csv file with same name, desired data type, size in seq etc

select \* from public.department; -- to see data is created with structure

use below command to load data

COPY PUBLIC.DEPARTMENT FROM 'C:\Users\pghare\Desktop\sample data set.csv' WITH CSV HEADER;

and then verify data using select \* from public.department; to check data loaded

------------------------------------------------------------------------------------------------------------

**18) Import xml file to Postgresql**

link: https://www.youtube.com/watch?v=JPn\_t60tGCw

a) Suppose we have xml file in some folder say Sales.xml having 1000 records data that we need to load

b) Open pgAdmin4 tool, and connect to postgresql db, check its version using select version() and check

its specific schema, existing tables there to verify details

c) Now in this query window, create the new table for importing data into it with simillar structure to xml file as below

CREATE TABLE IF NOT EXISTS public."Sellers"

(

id integer,

first\_name character varying(255) COLLATE pg\_catalog."default",

last\_name character varying(255) COLLATE pg\_catalog."default",

email character varying(255) COLLATE pg\_catalog."default",

gender character varying(255) COLLATE pg\_catalog."default"

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Sellers"

OWNER to postgres;

d) execute this script to get created our table and verify it

e) Now from the link https://www.postgresql.org/ftp/odbc/versions/msi/

download the latest version (x64 and x86 as compatible) ODBC drivers and install these two files in your pc

f) open windows administrative tools from run window, and select ODBC 32 bit driver from options, right click on it and open it

g) now we need to create user dsn, click on Add, select Postgresql Unicode and click on Finish

h) give the ODBC conn details like datasource, db name,server, port, user/pwd and test connection and save it and click ok

i) now open ssis package from VS, select integration services project, give new project name, location,

j) Now from the SSIS toolbox window, select Data flow task and XML source and drop them to right side window where we need to link them

k) confisure the xml source with file name etc, generate XSD, check columns and ok

l) drag ODBC destination from toolbox window to right side windows, connect xml source to it, edit this odbc destination window

using right click, create new conn, so click on new...

m) then on next screen, select your postgresql data source from dropdown , test conn, click ok. ok

n) in data access mode, select Table name - batch as we want to load data in batch, not like row by row

o) select the table name (dest table) as Sellers ( that we created earlier) from the drop down

p) click on mapping to see all mapping are good or update them as necessary, click ok

q) so our ssis package is ready, now execute it using STart button, and it will load all data into postgresql table

r) then you can verify data in postgresql Sellers table to complete this migration

------------------------------------------------------------------------------

**19) MYSQL Loading data from XML file into table**

a) Suppose we have one xml file related to EMPLOYEES data that we need to load into mysql table

b) create this new target table in mysql db schema say my\_schema with same structure as xml file , see sample below

CREATE TABLE 'my\_schema'.'emp'(

'emp\_no' INT NOT NULL,

'ename' VARCHAR(45) NULL,

'job' VARCHAR(45) NULL,

primary key ('emp\_no')

);

c) Now for this use below load\_data\_script.txt file

LOAD XML

INFILE "D:\MY\_FOLDER\emp\_data.xml"

INTO TABLE my\_schema.emp

ROWS IDENTIFIED BY '<EMPLOYEE>';

D) now open mysql workbench query window and use below queries

drop table if exists 'my\_schema'.'emp';

CREATE TABLE 'my\_schema'.'emp'(

'emp\_no' INT NOT NULL,

'ename' VARCHAR(45) NULL,

'job' VARCHAR(45) NULL,

primary key ('emp\_no')

);

select \* from my\_schema.emp;

e) Now connect to the mysql command line and execute below command to load data statement

mysql> LOAD XML

-> INFILE "D:\MY\_FOLDER\emp\_data.xml"

-> INTO TABLE my\_schema.emp

-> ROWS IDENTIFIED BY '<EMPLOYEE>';

and it will show how many affected or loaded into the dest table, then you can check data in mysql workbench window

by using command select \* from my\_schema.emp; to see data there to verify this dataload

-----------------------------------------------------------------------------------------------------

**20) Export data from Oracle database to excel file dynamically using Python Library: sample script**

# If the below two lib are not there, you may need to install it first in your env

# (Redhat Linux or windows etc using cmd # pip install cx\_Oracle )

# Download and install latest oracle instantclient library in your home directory from oracle download center

import cx\_Oracle

#import xlswt

import xlsxwriter

conn = cx\_Oracle.connect('SYSTEM/test@localhost/orcl.localdomain')

tab\_cur = con.cursor()

tab.cur.execute("Select distinct table\_name, count(column\_name) col\_count from user\_tab\_columns group by

table\_name order by table\_name")

#book = xlwt.Workbook(encoding=utf-8")

book = xlsxwriter.Workbook("export.xlsx")

for tab in tab\_cur:

#sheet1 = book.add\_sheet("Sheet 1")

sheet1 = book.add\_worksheet(tab[0])

col\_cur = con.cursor

col\_cur.execute("select distinct column\_name, column\_id from user\_tab\_columns where table\_name = '" + tab[0] + "' order by column\_id")

c=0

r=0

colname=""

print "Exporting table " + tab[0]

for col in col\_cur:

sheet1.write(r,c,col[0])

colname = colname+col[0])

if c<tab[1]-1:

colname=colname+','

c=c+1

col\_cur.close()

r=1

cur = con.cursor()

cur.execute("select " + colname + " from " + tab[0])

for row in cur

c=0

for i in range(tab[1]):

sheet1.write(r, c, row[i])

c=c+1

r=r+1

cur.close()

tab\_cur.close()

con.close()

book.close()

You can save this dynamic script with some name like export.py , can be used for any random no of rows and columns it has,

to export data in csv you need to fire command in redhat directory as: python export.py

and then go to your redhat documents directory and can get this csv file loaded with desired data

---------------------------------------------------------------------------------------------------

**21) FTP server client operation using Python**

# Using below scripts you can upload any type of file to FTP server using python FTP utility scripts

# you can check the file is uploaded or how and then read contents of it using below 3 scripts

putFile.py

import ftplib

import sys

filename = sys.argv[1]

connect = ftplib.FTP("Host IP address")

connect.login("oracle","oracle")

file=open(filename, "rb")

connect.storbinary("STOR " + filename,file)

connect.quit()

file.close()

getDir.py

import ftplib

connect = ftplib.FTP("Host IP")

connect.login("oracle","oracle")

data = []

connect.dir(data.append)

for line in data:

print(line)

readFile.py

import ftplib

import sys

filename = sys.argv[1]

connect = ftplib.FTP("Host IP")

connect.login("oracle","oracle")

connect.retrlines("RETR " + filename)

connect.quit()