

SOFTWARE LIBRE EN PYTHON PARA MIGRAR DE PARADOX EN WINDOWS A MYSQL EN LINUX, EN MODO BATCH

Fernando Velásquez Echeverry
Ingeniero mecánico.
Departamento Informática
Universidad Cooperativa Pereira
gitzu@hotmail.com

RESUMEN

La investigación busca construir un sistema, que sea de software libre, y que illustre y solucione el problema de migrar una tabla Paradox, en Windows, a otra en MySQL bajo Linux, de modo que pueda servir de base a una solución mucho mas general que involucre otros formatos de bases de datos. El procedimiento debe poder realizarse en batch para efectos de automatización.

Se tiene un sistema local con un sistema de bases de datos utilizados en las LAN (Local Area Network), como por ejemplo el formato de tablas Paradox utilizado por las herramientas de desarrollo de la compañía Borland: Delphi o Cbuilder, y se desea subir las tablas a un servidor Linux corriendo MySQL.

La solución se hace utilizando el lenguaje Python, que es un lenguaje de programación reciente, (www.python.org), muy reconocido a nivel internacional pero con una aparente poca difusión en Colombia.

Es un lenguaje de muy alto nivel, de los llamados tipo script, pero con características muy potentes que lo hacen también apto para aplicaciones complejas. Se eligió dadas sus características: Es software libre, es de alto nivel y es orientado a objetos.

El algoritmo utilizado es aplicable a otro tipo de migraciones, de modo que a partir de este proyecto de software se puede partir y desarrollar una mas general.

PALABRAS CLAVES

Python, MySQL, Paradox, Exportar, migración, ODBC

ABSTRACT

The investigation seeks to build a free software system, showing and resolving the

migration problem between Paradox in Windows and MySQL in Linux, creating a solution than make the base for a more general case, with several databases formats.

The process must be in batch mode, in order to automatize tasks.

Exist a local system with Paradox, working in a LAN and the objective is upload to MySQL, in Linux.

The answer is a program in Python, the new furor language, although no very popular in Colombia, yet.

This is a high level language, but with many power features, being used from simple OS scripts to complex applications.

It's free software, and object oriented. The alghoritm used is also targeted to another type of migrations, so starting from this project is possible to make one more general.

KEY WORDS

Python, MySQL, Paradox, Export, migration, ODBC

INTRODUCCION

Es posible realizar un programa de computador con Python que lea una tabla de Paradox y exporte los registros a una equivalente en MySQL. Existen productos comerciales que realizan ese tipo de conversiones e inclusive con mas formatos de bases de datos. Pero la potencia, elegancia y sencillez del Python son una motivación para realizar esta tarea de una manera un tanto pedagógica y como una propuesta base para hacer una utilidad migratoria mucho mas general.

Partiendo de la base que se conoce a Python, al iniciar se presentan varios desafíos como:

1. Que módulos se deben utilizar para tener soporte a las tablas Paradox?
2. Cuales para manipular las de MySQL?
3. Que API (Application Program Interface) debe utilizarse?
4. Que se debe tener en el entorno o en el equipo para realizar las pruebas correspondientes?
5. Que algoritmo aplicar para realizar la migración?

Se describe el contexto en el que se plantea el problema:

La aplicación exportadora está ubicada en el computador local, corriendo bajo el sistema operativo Windows, lo que implica que se va a desarrollar en Python para Windows. La herramienta que se utiliza en este caso es la que proporciona la empresa canadiense ActiveState (www.activestate.com), "Active Python", como un software gratuito descargable desde su página WEB.

CONTENIDO

BREVE INTRODUCCION A PYTHON

Python es un lenguaje de alto nivel de los llamados tipo script, relativamente nuevo y de propósito general. Su página oficial es www.python.org

Python es interactivo. Es un lenguaje interpretado y es orientado a objetos. Tiene tipos dinámicos y una sintaxis muy clara. El lenguaje es portable y corre en Unix, linux, Windows, Mac como mínimo.

MODULO O INTERFACE ESTANDAR PARA EL MANEJO DE BASES DE DATOS EN PYTHON

Existe un API (Application Program Interface) definido para que los diferentes módulos de bases de datos que existen en Python, tengan una uniformidad y una manera común de ser utilizados por parte del usuario, es decir, que independiente del módulo que se esté utilizando siempre sea con las mismos nombres de funciones que se realicen la conexión, los cursores etc., es decir, la misma interface. Los módulos que los desarrolladores de software hagan deben, por normalización, acogerse a este standard. Se utilizarán dos de estos modulos, uno el ODBC necesario para leer y

manipular, en este caso, las tablas paradox, y el módulo MySQL para manipular las tablas de MySQL.

MODULO ODBC

Con este módulo se puede acceder a las librerías ODBC instaladas en los equipos con sistemas operativos Windows.

Normalmente el ODBC tiene facilidades para acceder tablas de dBase, Foxpro, Access, Paradox etc.

MODULO MYSQL

Con este módulo se puede manipular las tablas MySQL, en nuestro caso, para insertar los registros que leamos de Paradox.

ALGORITMO DE EXPORTACION

Se debe recorrer la tabla parados, leyendo en cada registro los campos que la conforman e inmediatamente realizar una inserción en MySQL.

A continuación se presenta el código fuente. Este software se encuentra en producción en la sede Pereira de la Universidad Cooperativa de Colombia, y es utilizado para subir la información de Registro Académico a Internet. En el equipo en que está la información con las tablas Paradox, se instaló Python para ejecutar las instrucciones que siguen. Esta información ya migrada se puede consultar en la página web de la universidad, www.ucc.edu.co, escogiendo luego la ciudad de Pereira y finalmente seleccionando el vínculo Intranet.

MODULO PRINCIPAL

En este módulo se encuentran las funciones generales para que otra unidad de código las utilice. Es decir, es el módulo como tal.

```
import odbc,dbi
import MySQLdb
```

```
''' Módulo que exporta una tabla en Paradox a otra en mysql:
```

```
Se ha probado con campos de tipo integer y string.
```

```
No es necesario que tengan los mismos nombres de campo, pero si que tengan el mismo orden y sean de tipo compatible.
```

```
Debe convertirse a *.pyc para que no puedan leer las contraseñas del código
```

Debe tenerse en el equipo windows, en el módulo ODBC el driver de Paradox, nombrado como Paradox5.
 ""

Esta función es la responsable de construir la parte values de una instrucción SQL, debido a que estos valores cambian con cada registro insertado, se invocará tantas veces como registros se migren.

Recordar que la sintaxis de insert es: insert into TblMySQL (nombre, edad) values ("pepe",45), donde en este caso, los valores "pepe" y 45 son los que se van a leer de cada uno de los registros Paradox.

CurWin.description[i][1] es el equivalente a un arreglo en dos dimensiones donde i es el i-ésimo campo y 1 es el tipo del mismo. (0 es el nombre, para el lector curioso.)

Al finalizar esta función, retorna una cadena del estilo "pepe",45

```
def BuildValues(NumCampos, CurWin, RecWin):
    ""Construye la parte values del insert sql,
    que cambian con cada registro: ""
    temp2=""
    for i in range(0,NumCampos):
        #print CurWin.description[i][1]
        if CurWin.description[i][1]=='STRING': #
            El tipo del campo
            if RecWin[i] is None:
                ValorCampo=""
            else:
                ValorCampo=""+RecWin[i]+""
        elif
            CurWin.description[i][1]=='NUMBER':
            if RecWin[i] is None:
                ValorCampo=""
            else:
                ValorCampo=""+str(RecWin[i])+""
        else:
            ValorCampo=""
        if i!=NumCampos-1:
            temp2=temp2+ValorCampo+';'
        else:
            temp2= temp2+ValorCampo
    return temp2
```

Esta función es la principal del módulo, realiza las conexiones a las bases de datos en primer lugar. Si el parámetro append no es cierto (significa que se va a reemplazar

los registros en vez de añadir) se borran los registros de la tabla.

El paso siguiente es construir la primera parte de la instrucción insert con los nombres de los campos de la tabla MySQL.

Finalmente se seleccionan los registros de la tabla Paradox, y se recorre uno a uno utilizando la función insert que se ha construido dinámicamente.

Para finalizar se cierran las conexiones.

```
def Exporte(TblParadox,TblMySQL, pwd,
append,ipHost,usuario,baseDatos):
    try:
        ConnODBC= odbc.odbc('Paradox5') #
        podrá parametrizarse, igual que user etc.
        # Esta forma d pasar parámetros
        permite pasarlos en cualquier orden!!
        ConnMySQL =
        MySQLdb.connect(host=ipHost,
        user=usuario, passwd=pwd,db=baseDatos)
        CurWin = ConnODBC.cursor()
        CurMySQL = ConnMySQL.cursor()
        if not append:
            sql="delete from "+TblMySQL
            CurMySQL.execute(sql)
            sqlWin="select * from "+TblParadox
            CurWin.execute(sqlWin)
            #CurWin.description es una list
            describiendo el nombre de cada campo, su
            tipo etc.
            # se puede comprobar con print
            CurWin.description
            # necesitamos los nombres de los
            campos de la tabla en mysql y los valores de
            #los mismos que se toman de paradox
            sql="select * from "+TblMySQL
            CurMySQL.execute(sql) # necesario
            para averiguar NumCampos

            NumCampos=len(CurMySQL.description)
            temp=""
            # for necesario para construir la parte
            fields del insert sql:
            #insert into TblMySQL (nombre, edad)
            values ("pepe",45)
            for i in range(0,NumCampos):
                if i!=NumCampos-1:
                    temp=
                    temp+CurMySQL.description[i][0]+';' # [0]
                    tiene el nombre del campo
                else:
                    temp=
                    temp+CurMySQL.description[i][0]
            #print temp
```

```

#for tup in CurWin.description:
# print tup[0],
i=1;
while 1:
    RecWin = CurWin.fetchone()
    if not RecWin: break
    print "Registro: ",i,"\r",      # Para
imprimir en la misma posici□: \r,
    i += 1
    sql="insert into "+TblMySQL + " (" +
temp+      ") values      ("      +
BuildValues(NumCampos,
CurWin,RecWin)+")"
    #print sql
    CurMySQL.execute(sql)
CurWin.close()
ConnODBC.close()
CurWin=None
CurMySQL.close()
ConnMySQL.close()
ConnODBC=None
except NameError,e:
CurWin.close()
ConnODBC.close()
CurWin=None
CurMySQL.close()
ConnMySQL.close()
ConnODBC=None
print 'error ', e, 'undefined'

```

```

if __name__ == '__main__':
    "" Para probar este m□ulo: ""
    import paradox_Mysql, getpass

    TblParadox=raw_input("Tabla Paradox:")
    TblMySQL=raw_input("Tabla
MySQL(sensible a may□culas):")
    pwd=getpass.getpass("Password      de
acceso a la base de datos Mysql:")
    add=raw_input("Vacear primero la tabla
(s/n), s:")
    if add.upper()=='S' :
        add= True
    else:
        add=False

paradox_Mysql.Exporte(TblParadox,TblMyS
QL, pwd, add,"localhost","gitzu","gitzu")

```

MODULO QUE UTILIZA LAS FUNCIONES DEFINIDAS ANTERIORMENTE

Se puede ver aqui la simplicidad con que se utiliza el m□ulo de arriba.

Si se ignora el c□digo fuente anterior y se piensa en que es una caja negra, la sencillez y elegancia no puede ser mayor.

```

import os,paradox_Mysql, getpass

pwd=getpass.getpass("Password de acceso
a la base de datos Mysql:")
paradox_Mysql.Exporte(TblParadox='factdlle'
,TblMySQL='factdlle',pwd=pwd,

```

CONCLUSIONES Y RECOMENDACIONES

La migraci□n de bases de datos, con las modernas herramientas de software libre que existen hoy en d□a, nos permiten realizar proyectos de software interesantes, y que pueden personalizarse y acomodarse a nuestras necesidades. Los m□dulos de bases de datos incorporados o disponibles en Pyhon son supremamente poderosos y permiten operaciones inter-plataformas de una manera relativamente sencilla, y lo mas importante en procesos en batch, con la flexibilidad de hacerlos autom□ticos y sin intervenci□n de los usuarios., o visto de otra manera, se pueden utilizar, de manera transparente, cada cierto tiempo programandolos con el administrador de tareas de windows o con el cron de linux.

A partir del esquema presentado se pueden presentar soluciones en las que es necesario hacer desarrollos locales pero que posteriormente deban publicarse en un servidor WEB corriendo MySQL.

BIBLIOGRAFIA

ALONSO, JOSE MIGUEL. TCP/IP en Unix.
JOHN WORSLEY, JOSHUA DRAKE.

BRUCE ECKEL. Thinking in Python.

Manual de mysql. www.mysql.com

Manual y documentaci□n de Python.
www.python.org

MARK PILGRIM. Dive into python,
<http://diveintopython.org>

Practical PostgreSQL

SILBERSCHATZ, KORTH, SUDARCHAM.
Fundamentos de bases de datos. 4A ed.

