

**Examen 1.**

**Programación Integrativa.**

**Exportación e Importación de SQL a MongoDB con Python.**

**Maestra: Julia Patricia Sánchez Solís**

**Alan Leonardo Heredia Flores**

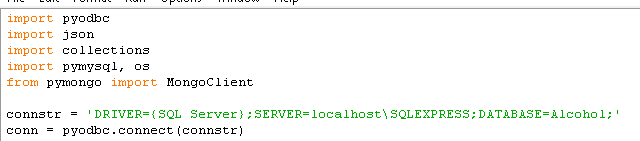
**Matricula: 117767**

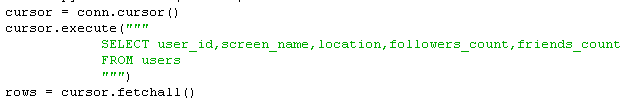
**Entrega: viernes 3 julio de 2020**

# Introducción.

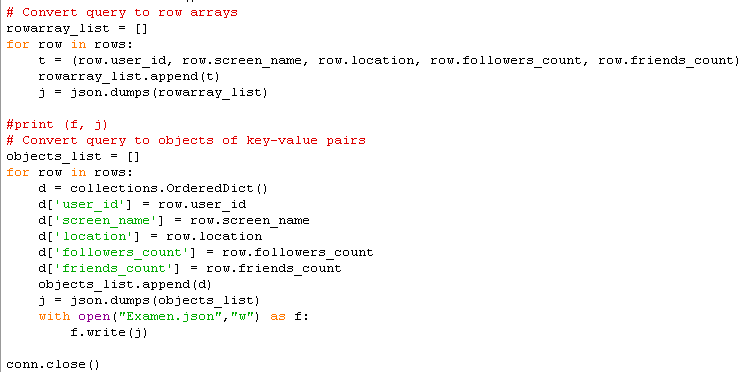
Este examen trata de cómo podemos migrar una tabla o una base de datos desde un gestor a otro con el uso un lenguaje de programación, en este caso exportare una tabla hecha en SQL Server, la cual se exportará con uso de Python y se importará automáticamente a MongoDB, después con ayuda de otro programa igual con Python podremos modificar los datos insertados en MongoDB y hacer las operaciones básicas CRUD (Inserción, Lectura, Actualización y Eliminación).

# Desarrollo.

1-. Primero necesitamos realizar la conexión a la base de datos que en este caso es SQL Server:

2-. Una vez realizada la conexión a la base, se realizará la consulta de esta, para localizar la tabla y los elementos de ella que se requieran para la exportación:

Se guardarán los datos obtenidos en la variable rows = cursor.fetchall().

3-. Una vez obtenidos los datos, se procederá a formar el archivo .json el cual será necesario para importarlo a el siguiente gestor de bases de datos, por lo que se utilizaran ciclos for para darle forma al archivo:

Una vez finalizada la extracción y formación del archivo .json se cierra la conexión al servidor de SQL.

4-.Una vez obtenido el archivo .json procederemos a realizar la conexión con MongoDB:

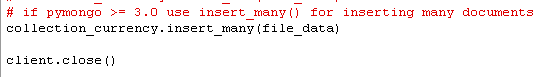


Aquí se declara el puerto, si es local y a base de datos y la colección en donde se insertarán los documentos.

5-. Una vez realizada la conexión y creación de la base y la colección, se tiene que leer el archivo .json creado anterior mente por lo que abriremos el archivo de la siguiente manera:



Se guardará la información del archivo .json en la variable file\_data.

6-. Una vez guardado el archivo en la variable file\_data, con una sentencia de mongo se insertará la información en la base de datos:

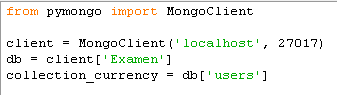
Aquí una observación con las sentencias de pymongo ya que hay diferentes entonces si son muchos documentos se usa insert\_many() en caso de que sea un documento nada más se utilizara insert\_one().

Y que esto es para la versión >= 3.0 de pymongo.

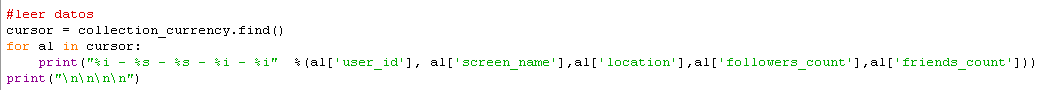
Con esto finalizamos la parte de migrar los datos de un gestor a otro, por lo que continuaremos con el CRUD de esta migración:

# CRUD básico MongoDB con Python.

1-. Primero realizaremos la conexión a la base MongoDB:



Seleccionaremos la base de datos y la colección en donde vamos a trabajar, después de que se conecta Python al servidor de MongoDB.

2-. Comenzamos con una lectura de la información de la colección a la cual estamos accediendo, mediante la consulta find() que se cuardara en la variable cursor, la cual vamos a utilizar en un ciclo for para recorrer todo el arreglo y en el cual vamos a imprimir parámetro por parámetro de los documentos:

#leer datos

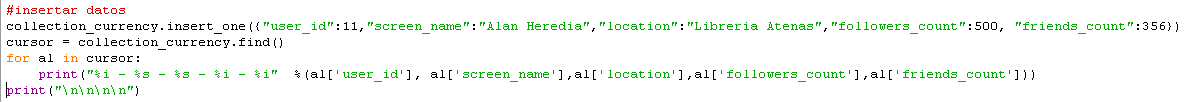
cursor = collection\_currency.find()

for al in cursor:

print("%i - %s - %s - %i - %i" %(al['user\_id'], al['screen\_name'], al['location'], al['followers\_count'], al['friends\_count']))

print("\n\n\n\n")

Podemos observar que para imprimir los valores de las claves del documento necesitamos declarar el tipo de variable ya sea int, string, float, etc… para que no tenga conflicto en reconocer los datos a la hora de leer e imprimir.

3-. Continuamos con una inserción de un documento en la misma colección con la siguiente consulta:

#insertar datos

collection\_currency.insert\_one({"user\_id":11,"screen\_name":"Alan Heredia","location":"Libreria Atenas","followers\_count":500, "friends\_count":356})

cursor = collection\_currency.find()

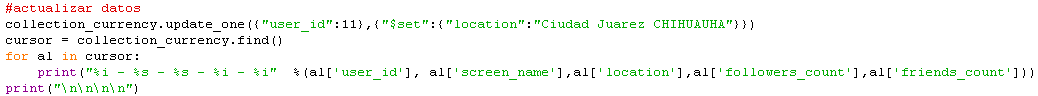
for al in cursor:

print("%i - %s - %s - %i - %i" %(al['user\_id'], al['screen\_name'],al['location'],al['followers\_count'],al['friends\_count']))

print("\n\n\n\n")

Podemos observar que para hacer la inserción, es igual que al insertar directamente a MongoDB desde el CMD, por lo que debemos tener cuidado en la sintaxis.

Al final mandamos a imprimir los datos para que se muestre el cambio y se diferencie de la primera consulta en donde estaban los documentos originales.

4-. Continuamos con la actualización de algún valor dentro de la colección, aquí podemos actualizar cualquiera de los valores de las variables de la colección, solo tenemos que especificar cual en la siguiente sentencia:

#actualizar datos

collection\_currency.update\_one({"user\_id":11},{"$set":{"location":"Ciudad Juarez CHIHUAUHA"}})

cursor = collection\_currency.find()

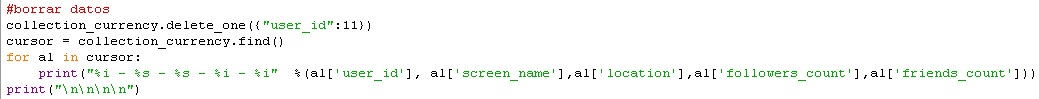
for al in cursor:

print("%i - %s - %s - %i - %i" %(al['user\_id'], al['screen\_name'],al['location'],al['followers\_count'],al['friends\_count']))

print("\n\n\n\n")

Como podemos observar necesitamos indicar primero en la consulta con update\_one() el atributo especifico de lo que queremos editar, puede ser con una variable existente, como en este caso lo fue el user\_id, pero podemos utilizar el id por defecto de la colección de mongo que da a sus documentos por ejemplo: {"\_id":ObjectId("5efc0569e5b318ac1b2c8b6e")}

Y cambiar cualquiera de las variables dentro del mismo documento, en este caso se cambio la locación del user\_id = 11 que se inserto en el paso anterior y que tenia como valor Ciudad Juares CHIHUAHUA. Después se imprime el contenido de la base para ver las diferencias.

5-. Por ultimo la sección de borrar un documento con la siguiente consulta:

#borrar datos

collection\_currency.delete\_one({"user\_id":11})

cursor = collection\_currency.find()

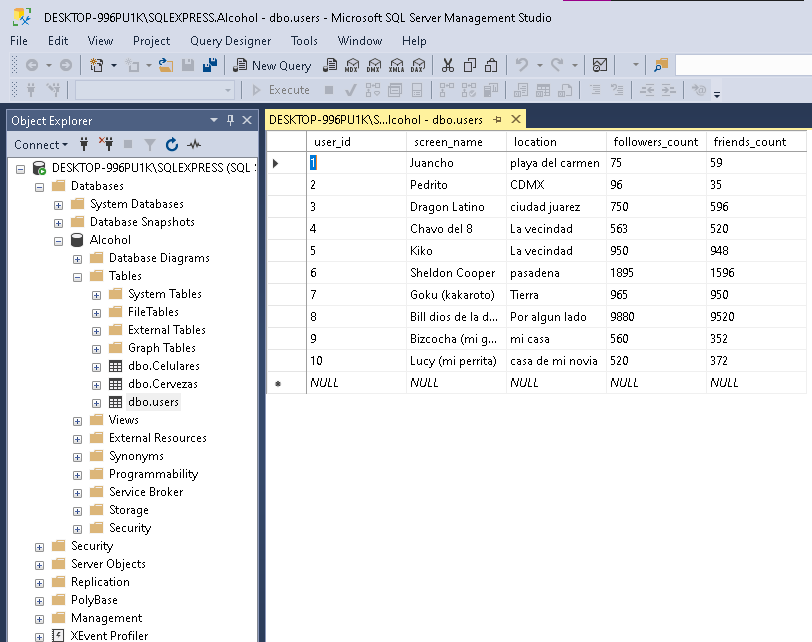
for al in cursor:

print("%i - %s - %s - %i - %i" %(al['user\_id'], al['screen\_name'],al['location'],al['followers\_count'],al['friends\_count']))

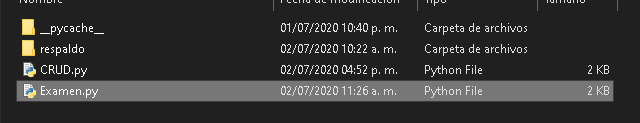
print("\n\n\n\n")

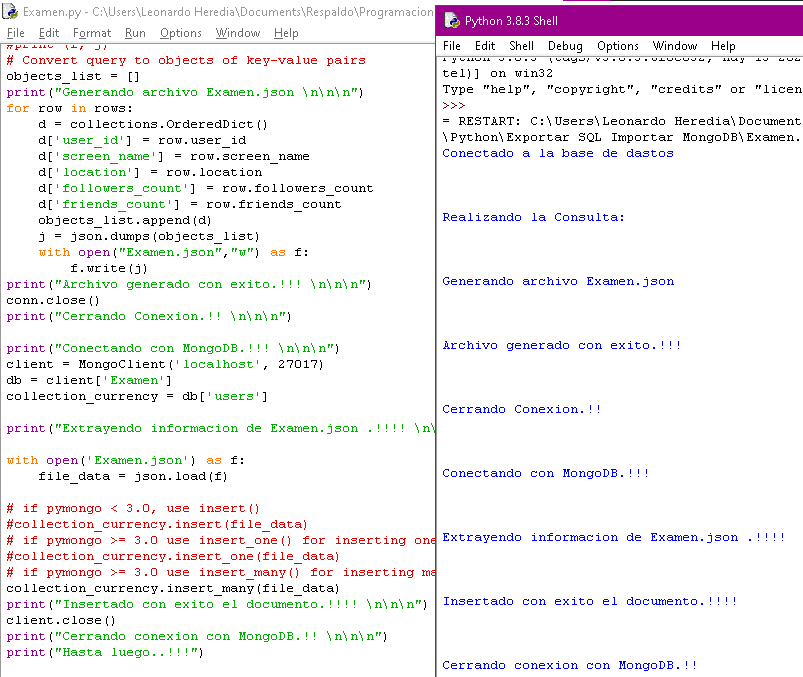
Podemos observar que para eliminar necesitamos delete\_one(), así eliminaremos completamente el documento y podremos imprimir el restante para ver la diferencia, podemos aquí mismo eliminar claves del documento, pero marca error ala hora de imprimir ya que falta un valor, aun así la clave es eliminada y puede verse reflejado directamente desde MongoDB en el CMD o en Mongo Compass.

# Capturas de pantalla del código ejecutándose.

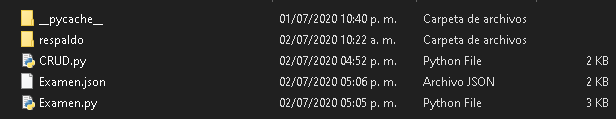
Esta es la base de datos en SQL Server que vamos a enviar a MongoDB.

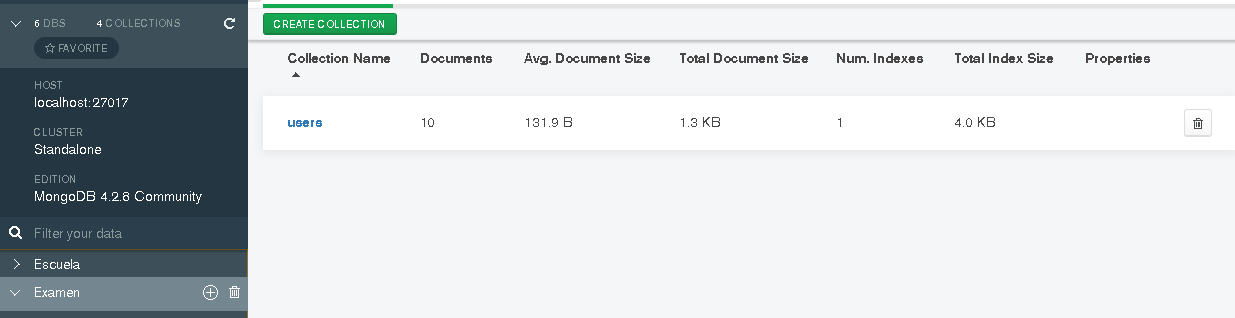
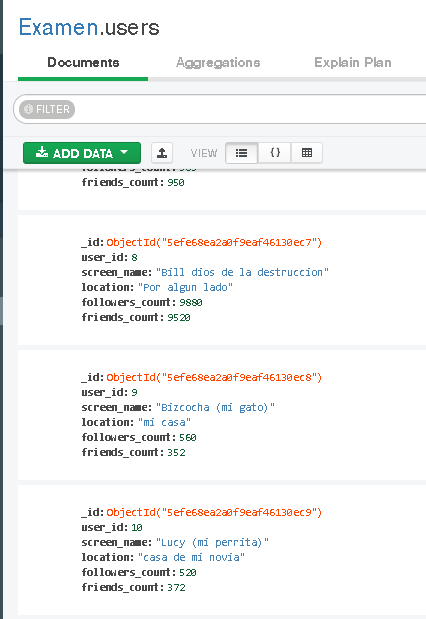
La carpeta que contiene el programa de exportación e importación esta sin el archivo .json, por lo que al correr el programa aparecerá el archivo Examen.json en esta misma carpeta.

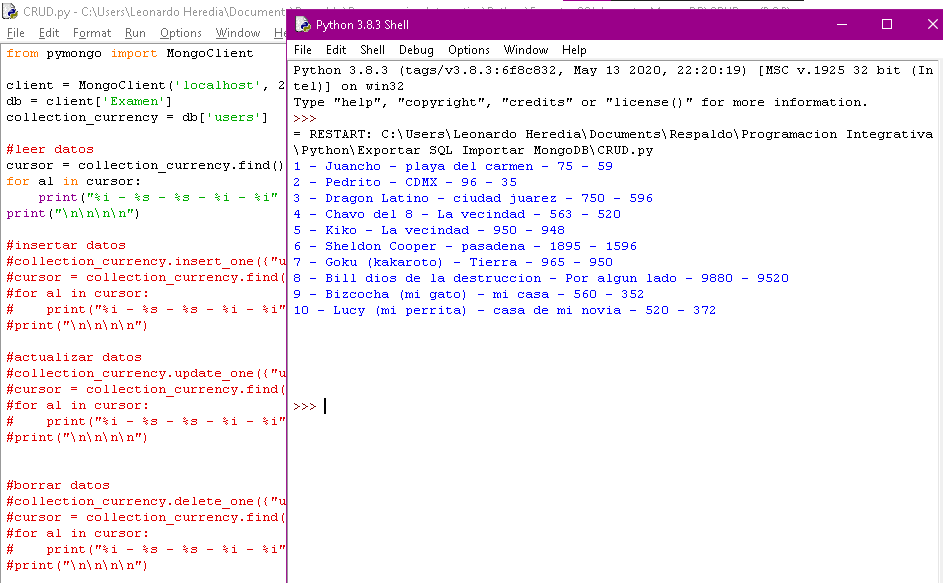


Ejecutando el programa de Python.

Ahora en la carpeta apareció el archivo Examen.json



En Mongo Compass aparece la base de datos.

Haremos la consulta de la base de datos desde Python.

Como podemos ver son 10 documentos, procederemos a insertar el documento 11, con los siguientes datos:

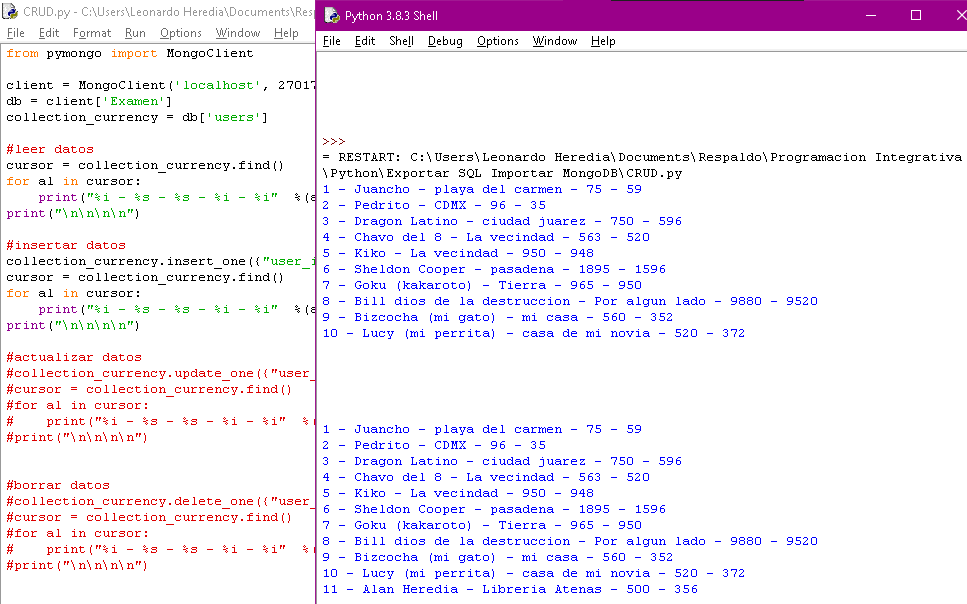
User\_id = 11

Screen\_name = ‘Alan Heredia’

Location = ‘Librería Atenas’

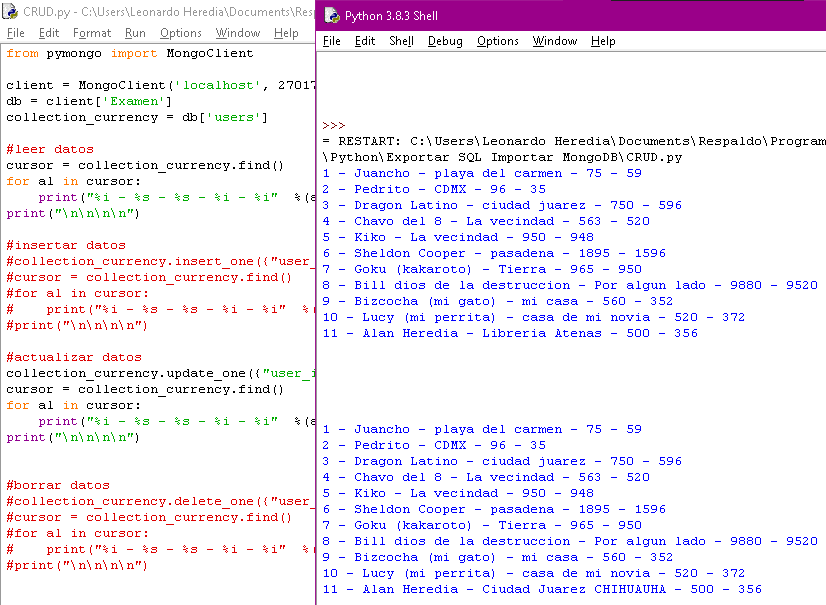
Followers\_count = 500

Friends\_count = 356

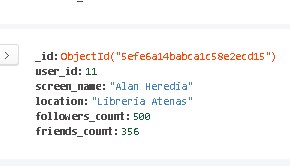
 Podemos observar que se inserto correctamente al final, y podemos demostrarlo con Mongo Compass.

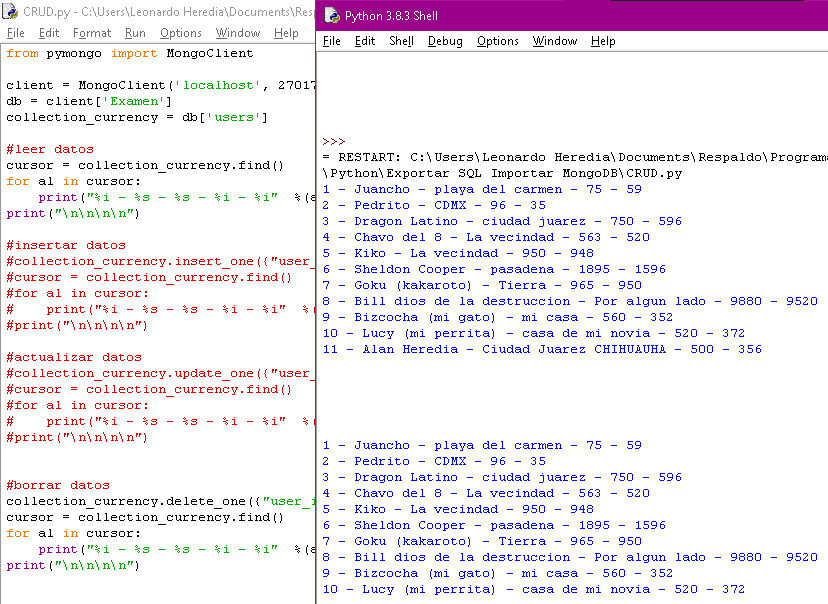


A continuación modificaremos la información de la locación del ultimo documento insertado, ósea el de user\_id= 11.

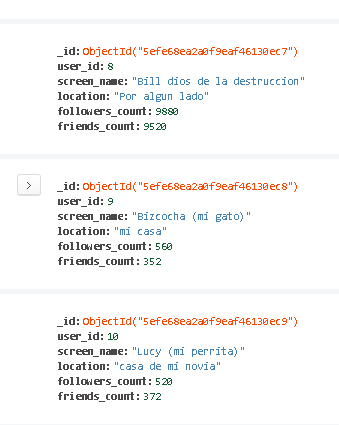
Cambiando la locación de Librería Atenas a Ciudad Juarez CHIHUAHUA.

Como podemos ver el cambio se realizo correctamente y podemos verificar en Mongo Compass.



Por ultimo pasaremos a eliminar un registro, por lo que seleccionaremos en el documento el user\_id = 11.

Y para verificar con Mongo Compass.



Esto seria todo por parte del documento y de mi examen, adjunto los programas y el documento.