How to handle bulk data insertion SQLite + python

When it comes of handling huge amount of data, the most common things that developer always does is to store data in a single manner each SQL statement has a new transaction started for it. This is very expensive, since it requires reopening, writing to, and closing the journal file for each statement. Despite that fact that they can do it in a bulk transaction. Now how do we did this? I'll show you.

Let's say you have 20,000 candidate records to be inserted in your database. It really makes sense to consider a bulk transaction right? Sure why not.

Convert Bytearray to Bytes in Python

Many different types of data objects are supported by Python. Two of them are the objects bytearray and bytes. The bytearray() function returns an array object of bytes. This object is changeable and supports the integer number from 0 to 255. The bytes() function returns bytes objects, is not changeable, and supports the integers from 0 to 255. This article will describe these functions and explain how bytearray objects can be converted into bytes objects.

List Intersection in Python

Many object variables exist in python to store a variety of data types. The list is one of these variables and can store different types of data for different needs. Sometimes, we need to find common, uncommon, or both common and uncommon data items from the multiple lists for
programming purposes. Python contains several built-in functions and operators that can
perform these types of tasks for Python sets. Finding common data from the multiple lists is
called list intersection, but there is no operator or built-in function for lists like sets to find the
common data items from multiple lists. This tutorial will show you how to intersect lists in
Python.

- **How to Execute Shell Commands in Python Using the Subprocess Run Method**[5]

  Subprocess is a built-in Python module that can be used to create new processes and interact
with their input and output data streams. In simpler terms, you can use it to run shell
commands and run executable binaries usually scattered in various ?bin? folders across a
Linux file system. You can also supply a full path to an executable binary and use any
command-line switches associated with the binary. This article will explain how to use the
subprocess module and its run method in Python apps. All code samples in the article are
tested with Python 3.8.2 on Ubuntu 20.04.

- **How to Use the Python Isalpha Function**[6]

  Sometimes, we need to check the content of data for programming purposes. There are many
different types of built-in functions in Python for string data to check the content. This content
may include letters, numbers, or other special characters. The isalpha() function is one of the
useful built-in functions of Python that can be used to find out whether or not the content of
the data is alphabetic. This function searches the alphabet in the starting of the string value. If
the starting value of the string is a letter, then this function returns true; otherwise, it returns
false. This tutorial will show you how to can use the isalpha() function in Python.

- **PSF GSoC students blogs: Weekly Blog #1**[7]

  Hello Everyone, this is Soham Biswas currently in 2nd year pursuing my Bachelor’s(B.Tech)
degree in Computer Science & Engineering from Institute of Engineering & Management,
Kolkata. I have been selected for GSoC’ 20 at sub-org FURY under the umbrella organisation
of Python Software Foundation. I will be working on building sci-fi-like 2D and 3D interfaces
and provide physics engine integration under project titled "Create new UI widgets & Physics
Engine Integration".

- **PSF GSoC students blogs: First Blog GSoC 2020**[8]
Since most of the places including my university are closed due to the pandemic outbreak, I decided to get a head start and start with the project early. During the community bonding period, I had video conference meetings with my mentors scheduled every week on Wednesday. During these meetings I interacted with the mentors to have a coherent understanding of how the project design and implementation will be managed over the course of the entire period.

Since my project involves a lot of theoretical understanding of concepts such as ray marching, I spent the period going through the theory of each topic. This week also involved going through the documentation for shaders used in VTK.

I had an onboarding meeting with my mentors where we got to know each other a bit better. They advised me to play around with uarray and unumpy without any goal in mind which I found to be a very good advice. I played a bit with special methods by implementing a simple Vector2D class and used the code in this notebook with some print statements to understand better the protocols and how they are called. I wanted to start earlier on my project so I took over a PR from one of my mentors which adds multimethods for the linalg module.

**What is coming up next?**

I'm going to continue the PR that I have been working on since it still isn't finished and I will also follow the proposed timeline and start adding multimethods for other routines like checking class equality in array elements. Some mathematical constants and their aliases are also missing so I will be adding these too and probably refactoring the existing ones into classes. This week marks the end of my college classes but I still have some assignments and exams coming up in the following weeks so there's a lot of work ahead of me to properly balance both university studies and GSoC but I wouldn't have it other way.