

Project Name: SQL Python Sqlite3

Table of Contents

Demo
Overview
Motivation
Technical Aspect
Installation
Run/How to Use/Steps
Directory Tree/Structure of Project
To Do/Future Scope
Technologies Used/System Requirements/Tech Stack
Credits

Demo

WAREHOUSE INVENTORY SALES PURCHASE MANAGEMENT SYSTEM

Warehouse Inventory Sales Purchase

Product Item Details:

Product Id:	12
Product Name:	Wire
Product Price:	20
Product Quantity:	5
Product Mfg Company Name:	Havells
Company Contact Number:	9876541290

Product Item Information:

12 {Wire } 5 20 Havells 9876541290

Save Data

Show Data

Reset

Delete

Search

Update

Close

```
Anaconda Prompt (anaconda3) - python 1)WarehouseInventorySalesPurchase_Python_with_sqlite3.py
(base) C:\Users\Monica>cd C:\Users\Monica\Desktop\Projects\Python Projects 1\18)Python _with_SQL\Project1 python+sqlite3
(base) C:\Users\Monica\Desktop\Projects\Python Projects 1\18)Python _with_SQL\Project1 python+sqlite3>python 1)Warehouse
InventorySalesPurchase_Python_with_sqlite3.py
Database : connection method called
Database : connection method finished

Product : showInProductList method called
Database : show method called
Database : show method finished
```

This is diving into SQL Concept along with Python using sqlite3.

SQL is used to communicate with a database. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database, which is called CRUD Operations. This repository contains the code for SQL with Python using python's various libraries.

It used tkinter and sqlite3 libraries.

These libraries help to perform individually one particular functionality.

Python when combined with Tkinter library provides a fast and easy way and is used to create GUI applications.

Tkinter is the inbuilt python module that is used to create GUI applications.

SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.

The purpose of creating this repository is to gain insights into working with Queries in Python.

These python libraries raised knowledge in discovering these libraries with practical use of it.

It leads to growth in my ML repository.

This above few screenshots will help you to understand flow of output.

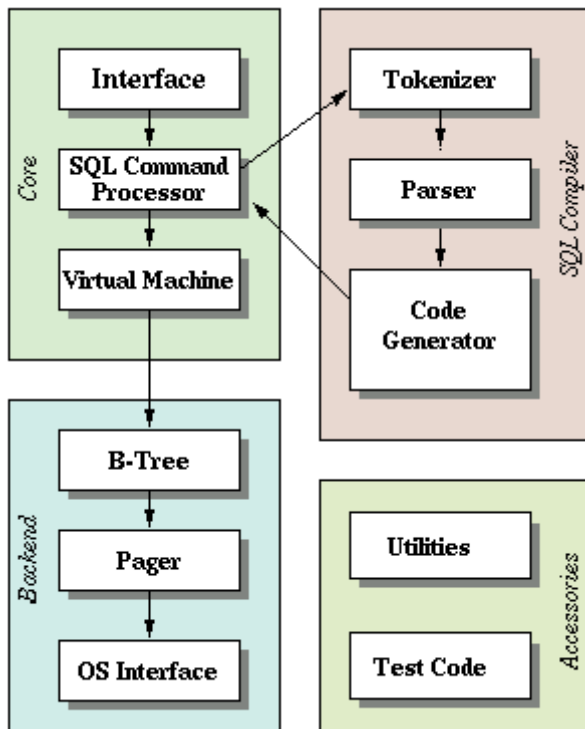
Motivation

The reason behind building this is, I wanted to work with combination of database with python integration. So, SQLite is capable of creating in-memory databases that are very fast to work with. Other point that caught my eye is, SQLite has higher performance. SQLite databases can also be queried and the data retrieval is much more robust. I have already learnt SQL subject in my graduation years so this is an easy task to start with for me. SQLite supports transactions and atomic behaviours, so a program crash or even a power outage won't leave you with a corrupted database. As a Data Scientist, working with database and collecting and transferring data from database is a routine work so it is an essential for me get hands on. A database is typically designed so that it is easy to store and access information. A good database is crucial to any company or organisation. This is because the database stores all the pertinent details about the company such as employee records, transactional records, salary details etc. Hence, I continue to gain knowledge while practicing the same and spread literary wings in tech-heaven.

Technical Aspect

Tkinter provides handy to use flow for anyone who is using it. It gives a powerful object-oriented interface to the Tk GUI toolkit. You don't need to worry about the installation of the Tkinter module separately as it comes with Python already.

SQLite is a server-less database and is self-contained. This is also referred to as an embedded database which means the DB engine runs as a part of the app. On the other hand, MySQL requires a server to run. MySQL will require a client and server architecture to interact over a network.



Installation

Using intel core i5 9th generation with NVIDIA GFORCE GTX1650.

Windows 10 Environment Used.

Already Installed Anaconda Navigator for Python 3.x

The Code is written in Python 3.8.

If you don't have Python installed then please install Anaconda Navigator from its official site.

If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python -m pip install --upgrade pip and press Enter.*

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.

Follow this when you want to perform from scratch.

Install SQLite3.

Open Anaconda Prompt, Perform the following steps:

`cd <PATH>`

`pip install sqlite3`

You can also create requirement.txt file as, `pip freeze > requirements.txt`
run files.

Follow this when you want to just perform on local machine.

Download ZIP File.

Right-Click on ZIP file in download section and select Extract file option, which will unzip file.

Move unzip folder to desired folder/location be it D drive or desktop etc.

Open Anaconda Prompt, write `cd <PATH>` and press Enter.

eg: `cd C:\Users\Monica\Desktop\Projects\Python Projects 1\18)Python_with_SQL\Project1`
`python+sqlite3`

In Anconda Prompt, `pip install -r requirements.txt` to install all packages.

In Anconda Prompt, write `<filename>.py` and press Enter. That is,

In Anconda Prompt, write `1)WarehouseInventorySalesPurchase_Python_with_sqlite3.py`

This creates `inventory.db` file as output and then I created empty folder named `Output_Files` and transferred it there.

Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.

Note: `cd <PATH>`

[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to `cd` one space `<path>` and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

Folder: `18)Python_with_SQL > Project1 python+sqlite3`

`1)WarehouseInventorySalesPurchase_Python_with_sqlite3.py`

To Do/Future Scope

Can work with other type of GUI applications.

Technologies Used/System Requirements/Tech Stack



Credits

tech fort channel