Topic Challenge - Module 6D - Understanding Database Options

Instructions:

Your boss calls you into a meeting. She tells you about a new project that is about to begin. It needs a database back-end, and she would like you to do some research to figure out which database is the best solution. She says that the first phase of the project is just to test feasibility, so they need a very simple, fast solution that doesn't need a lot of setup or special hardware/software to get going. It needs to be cross platform but doesn't require any special network access. She says for the feasibility phase, only one user will be accessing the database from their computer. She says it needs to be easy to backup to a usb stick so it can be copied from one computer to another. Cost is of utmost importance during this first phase of this project. It's ok if the database backend is replaced with something else in later phases, once all the requirements have been worked out. Oh, and one last thing, it needs to be compatible with Python, since that's what will be used throughout this project.

Given the above information, make a list of features (criteria) that are important to your boss for this phase of the project. Rank how important each criteria is, where 1 is not very important, and 5 is very important. Show your criteria and the rank you assigned each one.

Evaluate the following databases against your criteria:

- MSSQL
- Oracle
- SQLite
- MySQL (or MariaDB)
- PostgreSQL
- Microsoft Access
- LibreOffice Base

Rate each database against the criteria, where 0 means it doesn't meet the criteria at all, and 5 meets the criteria extremely well. Show your rating for each database, against each criteria.

Multiply your numbers and add them up, just like the class example. Find which database should work the best for this phase of the project, and state your findings.

Criteria	MSSQL	Oracle	SQLite	MySQL	PostgreSQL	Microsoft Access	LibreOffice Base
Simplicity & ease of setup	2	2	5	4	4	3	3
Speed & Performance	4	4	5	4	4	3	3
Cross-platform compatibility	1	1	5	4	5	1	5
Single user access	5	5	5	5	5	5	5
Easy backup and portability	1	1	5	4	5	3	4
Cost-effectiveness	2	2	5	4	4	3	5
Compatibility with Python	1	1	5	5	5	2	4
Total	16	16	35	30	32	20	29
	52	52	175	130	148	66	125

SQLite is better for this project because it's:

- **Simple:** Easy to set up and use, with no server setup required.
- **Fast:** Optimized for single-user scenarios, providing efficient performance.
- **Cross-platform:** Works seamlessly across different operating systems.
- **Single-user focused:** Perfect for scenarios with only one user accessing the database.
- **Portable:** Databases are stored in a single file, making them easy to backup and transfer.
- **Cost-effective:** Open-source and free to use.
- **Python-compatible:** Integrates well with Python applications.

Reasons for the lower scores of alternative options:

Oracle and MSSQL:

- Complexity: Setting up and maintaining an Oracle or MSSQL database requires specialized servers and a great deal of configuration.
- Cost: Because MSSQL and Oracle are private databases that require licensing, they are not as economical for applications with tight budgets.

MariaDB, also known as MySQL:

- Requires less setup and maintenance; yet it is a well-liked open-source alternative.
- Cost: Although MySQL/MariaDB is open source, its cost-effectiveness may be impacted by the additional resources needed for setup and upkeep.

PostgreSQL:

- It could be too complex for the earliest stages of a project.
- It requires more setup and management.

LibreOffice Base and Microsoft Access:

- Cross-platform interoperability is limited by Microsoft Access's Windows platform dependence.
- Despite being cross-platform, LibreOffice Base might not provide the functionality and performance required for larger-scale applications.