**CONSTRAINTS**

**First phase of the project (Test Feasibility)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Constraints** | **Rate** | **MS SQL** | **Oracle** | **SQL Lite** | **MYSQL (MariaDB)** | **PostgreSQL** | **Microsoft Access** | **LibreOffice Base** |
| **Ease of Installation** | 5 | 2 | 2 | 5 | 4 | 3 | 1 | 1 |
| **Minimal Hardware Requirements** | 5 | 2 | 1 | 5 | 4 | 4 | 3 | 3 |
| **Low Maintenance (back-up)** | 4 | 2 | 1 | 4 | 3 | 2 | 2 | 2 |
| **Compatibility with other platform** | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 |
| **Cost-Effectiveness** | 4 | 2 | 1 | 4 | 3 | 3 | 2 | 2 |
| **Compatibility with Python** | 3 | 2 | 2 | 3 | 3 | 3 | 1 | 1 |
| **Scalability (Can change DB later)** | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
|  |  | **47** | **34** | **93** | **75** | **67** | **44** | **44** |
|  |  | **12%** | **8%** | **23%** | **19%** | **17%** | **11%** | **11%** |

**Find which database should work the best for this phase of the project and state your findings.**

SQLite emerges as a strong contender for this project scoring 23% this is due to its simplicity, minimal setup requirements, cross-platform compatibility, cost-effectiveness, and compatibility with Python. Its serverless architecture, along with the ease of integration with Python, makes it an attractive option for projects with limited resources or where scalability is not a primary concern. Additionally, SQL lite may not be as scalable as client-server databases, but it is well-suited for small to medium-sized projects or applications with a limited user base. I believe this is what the boss needed.