**SQLite vs Firebase**

|  |  |  |
| --- | --- | --- |
|  | **PRO** | **CONS** |
| SQLite | * SQL database => portable * Serverless * Queries are smaller than equivalent procedural codes => chances of bugs are minimal * Accessible with no installation * Cross-platform database | * Doesn’t provide network access * Not built for large-scale applications * Restricted to 2GB in most cases * No documentation * No user management |
| Firebase | * Free start * Concise documentation * Accessible UI and ease of integration * Real-Time data updates * Stored in the cloud => readily available everywhere * Serverless | * Limited to 100 Connections and 1GB of Storage * NoSQL database => hard to migrate * Very limited querying and indexing * Can’t query or list users or stored files |

**SQL vs NoSQL**

|  |  |  |
| --- | --- | --- |
|  | **PRO** | **CONS** |
| SQL | * Relational database * Favors normalized schema. * Well- Defined Standards * Data Integration Scripts * Code-free nature => hassle-free process | * Complex structure => it becomes difficult for certain users to access it * The process of interfacing is complex * It provides only partial control * The expenses involved in SQL operation is too much => it becomes difficult to bring vendor-in |
| NoSQL | * Flexible data model * Open source; low cost * Elastic scalability * Non-relational or distributed databases | * Almost no support * No standardized platform * No interfaces and interoperability |

Link git: <https://github.com/MariaBrob/dapm/tree/master/dapm_lab8>