Creating Own Database and Compare Performance with SQLite

Team Members:

|  |  |  |
| --- | --- | --- |
| Ojasva Jain | Section – A (43) | 140905490 |
| Aayush Gupta | Section – A (45) | 140905518 |

Abstract:

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is lightweight, fast, and compact. And it works completely out of the box without any configuration. SQLite is the most widely deployed database engine in the world today. A few of the better known users of SQLite are Adobe, Google, Microsoft and Apple.

While SQLite is very fast at first, it is disliked due to its inefficient use of space, and it has a dramatically slow down overtime when a lot of data is being written and/or deleted. SQLite also mixes data and indexes into the same file, making linear iteration a tedious job, especially when records get deleted, it leaves gaps that never get filled until vacuum() command is executed. So, we want to make a database that is just as fast as SQLite, but does not slow down overtime, better reusing of deleted space and does not require vacuum().

The main goal of this project is to implement our own database, query from the database, display the results of the query and to learn how database works. We will then compare the performance of our database with SQLite.

Gantt Chart:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Team Member | Week 1 | Week 2 | Week 3 | | Week 4 | Week 5 | | Week 6 | Week 7 | Week 8 | | Week 9 | | Week 10 |
| Aayush | Block Storage Design, Indexing | Block Storage Implementation | | Record Storage Implementation | | | B-Tree Implementation | | | | Creating Database | | Comparing Database | |
| Ojasva | Record Storage Design, Query Parsing |

Mid Term Deliverables:

* Design of the Database
  + Block Storage
  + Record Storage
  + Indexing
  + Query Parsing
* Implementation
  + Block Storage Implementaion
  + Record Storage Implementation

