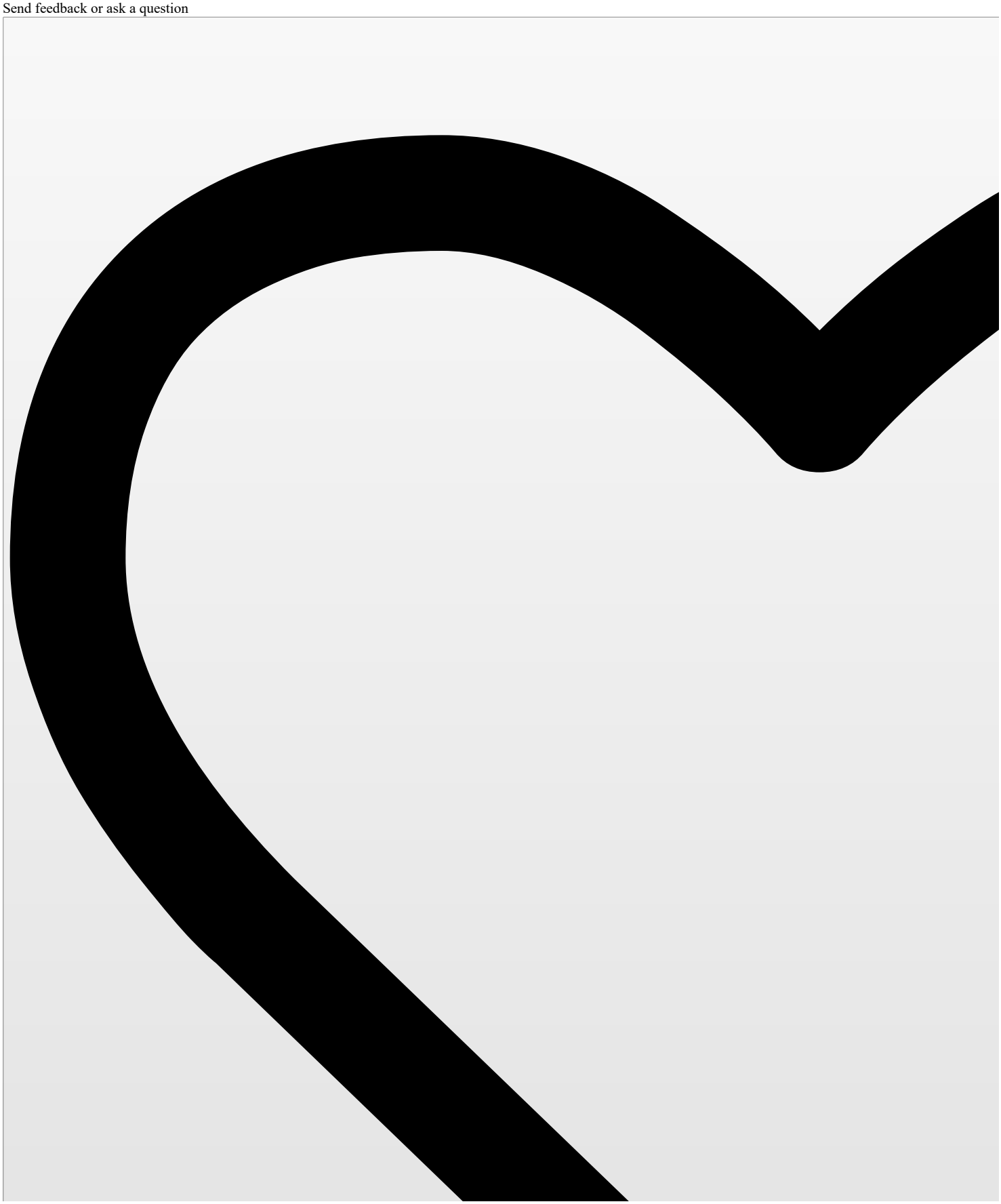
Indexes

Indexes are well known when it comes to databases; they are used to improve the speed of data retrieval operations on the data store. An index makes the trade-offs of increased storage overhead, and slower writes (since we not only have to write the data but also have to update the index) for the benefit of faster reads. Indexes are used to quickly locate data without having to examine every row in a database table. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.

An index is a data structure that can be perceived as a table of contents that points us to the location where actual data lives. So when we create an index on a column of a table, we store that column and a pointer to the whole row in the index. Indexes are also used to create different views of the same data. For large data sets, this is an excellent way to specify different filters or sorting schemes without resorting to creating multiple additional copies of the data.

Just as to a traditional relational data store, we can also apply this concept to larger data sets. The trick with indexes is that we must carefully consider how users will access the data. In the case of data sets that are many TBs in size but with very small payloads (e.g., 1 KB), indexes are a necessity for optimizing data access. Finding a small payload in such a large data set can be a real challenge since we can't possibly iterate over that much data in any reasonable time. Furthermore, it is very likely that such a large data set is spread over several physical devices—this means we need some way to find the correct physical location of the desired data. Indexes are the best way to do this.

Mark as completed
← *Previous* Sharding or Data Partitioning *Next* → Proxies
Send feedback or ask a question



20 recommendations

- Home
- Featured
- Team
- Blog
- FAQ
- Terms of Service
- Contact Us