Best Practices to keep in mind, while creating a database to increase performance and scalability.

1. Data Partitioning: Divide the data into smaller, manageable chunks to distribute across multiple nodes, which improves query performance and parallel processing capabilities.
2. Indexing: Implement appropriate indexing techniques to speed up data retrieval and reduce search time. Choose between different index types like bitmap, hash, and B-tree, based on the specific use case.
3. Data Compression: Apply data compression techniques to reduce storage space and improve query performance. This can also minimize I/O operations and network latency.
4. Caching: Utilize caching mechanisms to store frequently accessed data in memory, reducing the need for disk I/O and improving overall performance.
5. Query Optimization: Optimize queries by leveraging techniques like predicate pushdown, join reordering, and query rewriting. Use a query optimizer to identify and eliminate performance bottlenecks.
6. Schema Design: Design the database schema with performance in mind, using denormalization and materialized views when necessary to improve read performance.
7. Data Storage Format: Choose an appropriate data storage format like columnar or row-based, depending on the analytical requirements and access patterns.
8. Scalability: Design the database to scale horizontally by adding more nodes, allowing it to handle increasing data volume and processing demands efficiently.
9. Load Balancing: Implement load balancing techniques to distribute workload evenly across nodes, preventing performance bottlenecks and ensuring high availability.
10. Monitoring and Tuning: Continuously monitor the database's performance, identify issues, and fine-tune various components, such as hardware, operating system, and database configuration, for optimal performance.