Arslan Ahmad • Following

Author of Bestselling 'Grokking' Series on System Design, Software Architecture & Coding ...

1d •

How to improve database performance?

Here are some of the top ways to improve database performance:

1. Indexing

Create the right indexes based on query patterns to speed up data retrieval.

2. Materialized Views

Store pre-computed query results for quick access, reducing the need to process complex queries repeatedly.

3. Vertical Scaling

Increase the capacity of the hashtag#database server by adding more CPU, RAM, or storage.

4. Denormalization

Reduce complex joins by restructuring data, which can improve query performance.

5. Database Caching

Store frequently accessed data in a faster storage layer to reduce load on the database.

6. Replication

Create copies of the primary database on different servers to distribute read load and enhance availability.

7. Sharding

Divide the database into smaller, manageable pieces, or shards, to distribute load and improve performance.

8. Partitioning

Split large tables into smaller, more manageable pieces to improve query performance and maintenance.

9. Query Optimization

Rewrite and fine-tune queries to execute more efficiently.

10. Use of Appropriate Data Types

Select the most efficient data types for each column to save space and speed up processing.

11. Limiting Indexes

Avoid excessive indexing, which can slow down write operations.

12. Archiving Old Data

Move infrequently accessed data to an archive to keep the active database smaller and faster.

📌 Grokking the System Design Interview - **https://lnkd.in/giwyzfkT**

📌 Liked this post? Join my free newsletter: **https://lnkd.in/gpHAFd9t**

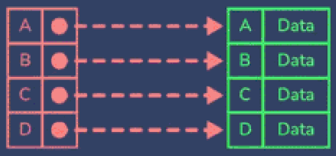
#systemdesign #databases #InterviewTips #SoftwareDevelopment

How to improve database performance?

{≡} DesignGurus.io

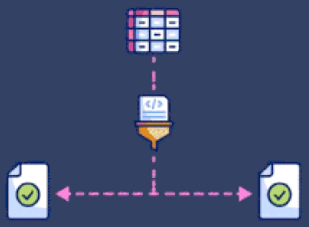
1. Indexing

Create the right indexes based on query patterns to speed up data retrieval.



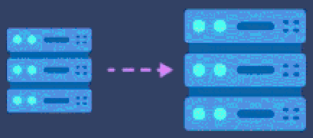
2. Materialized Views

Store pre-computed query results for quick access, reducing the need to process complex queries repeatedly.



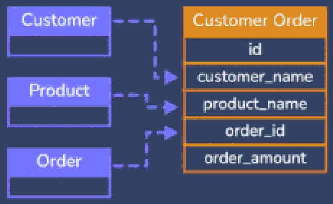
3. Vertical Scaling

Increase the capacity of the database server by adding more CPU, RAM, or storage.



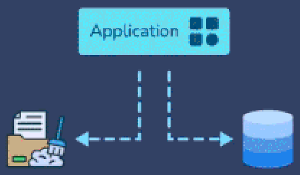
4. Denormalization

Reduce complex joins by restructuring data, which can improve query performance.



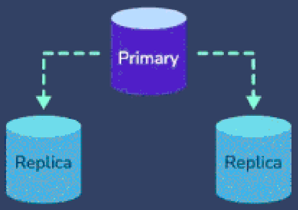
5. Database Caching

Store frequently accessed data in a faster storage layer to reduce load on the database.



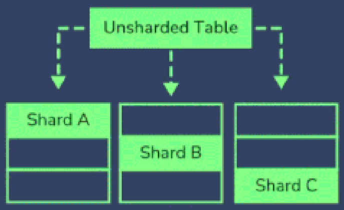
6. Replication

Create copies of the primary database on different servers to distribute read load and enhance availability.



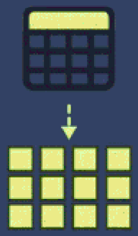
7. Sharding

Divide the database into smaller, manageable pieces, or shards, to distribute load and improve performance.



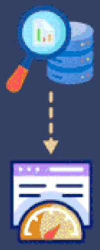
8. Partitioning

Split large tables into smaller, more manageable pieces to improve query performance and maintenance.



9. Query Optimization

Rewrite and fine-tune queries to execute more efficiently.



400 21 comments 94 reposts