**Horizontal Partitioning (Sharding):**

* **Concept**: In horizontal partitioning, the rows of a table are divided across multiple tables or databases. Each partition (or shard) holds a subset of the rows but the same columns as the original table.
* **When to Use**: It's used to improve performance and manageability when dealing with a large number of rows.

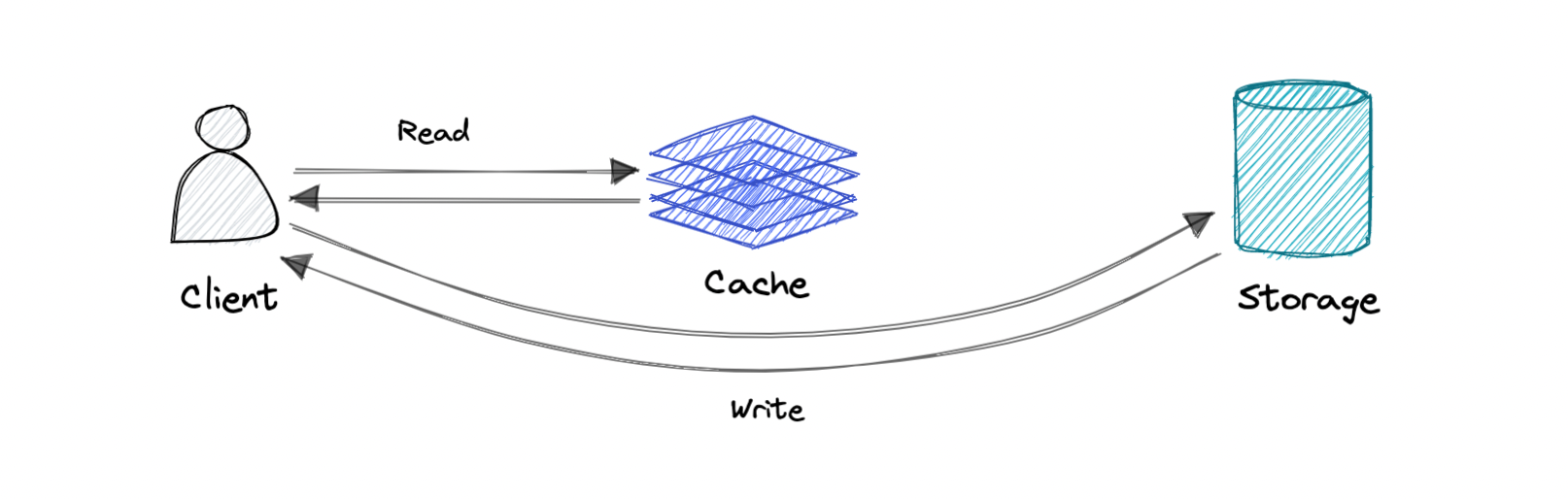
**Vertical Partitioning:**

* **Concept**: In vertical partitioning, the columns of a table are divided into multiple tables. Each partition holds a subset of the columns and all rows.
* **When to Use**: Useful when a table has many columns, and queries often access only a subset of those columns. It helps in reducing I/O by loading only the necessary columns.

**Caching stuff here**

**Cache miss?**  
A cache miss refers to the instance when the memory is searched, and the data isn't found.

**Write-around cache**

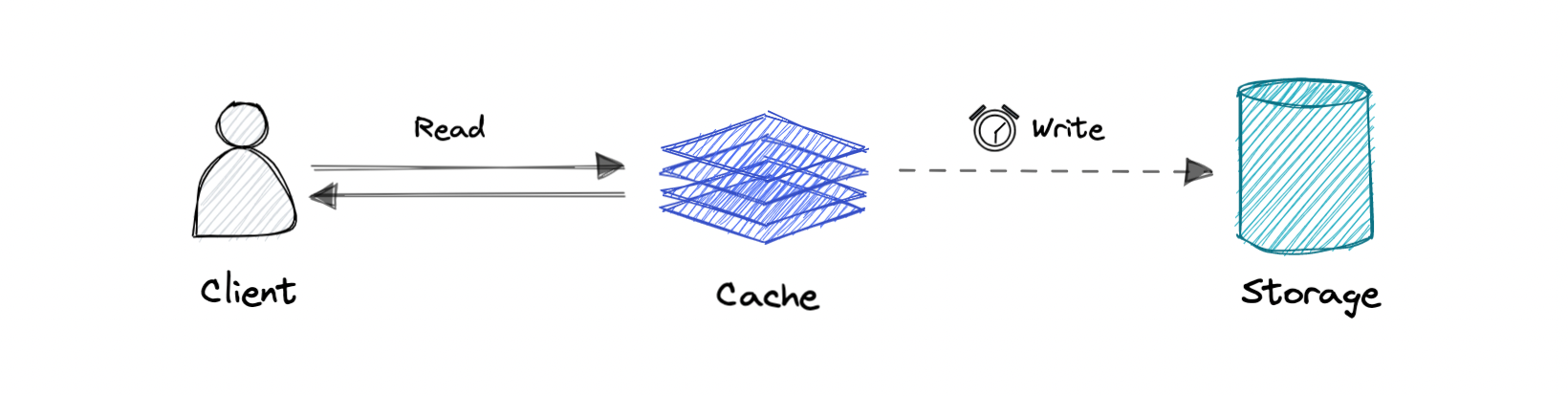


Where write directly goes to the database or permanent storage, bypassing the cache.

**Pro**: This may reduce latency.

**Con**: It increases cache misses because the cache system has to read the information from the database in case of a cache miss. As a result, this can lead to higher read latency in the case of applications that write and re-read the information quickly. Read happen from slower back-end storage and experiences higher latency.

**Write-back cache**



Where the write is only done to the caching layer and the write is confirmed as soon as the write to the cache completes. The cache then asynchronously syncs this write to the database.

**Pro**: This would lead to reduced latency and high throughput for write-intensive applications.