Caching is the process of temporarily storing the data at a certain location, such that you can use the data multiple times without directly reading from the databases, and thus enhancing the application’s performance. Caching helps to reduce the number of calls that are made to the database.

The data that is constantly changing is not a good candidate for caching, whereas the data that does not change very frequently is a good candidate for caching.

Cache-Write

Write-Through: The data is updated in the cache and the database (DB) simultaneously. The data in the cache and the data in the DB are always in sync.

Write-Back: The data is updated only in the cache and then updated in the DB at a later stage, using asynchronous calls. This strategy is also known as Write-Deferred.

Write-Around: Data is uploaded first into the DB and is later loaded in the cache.

Cache Eviction

FIFO: The item that enters the cache first is evicted first, irrespective of how often or how long ago it was accessed.

LIFO: The most recent item is evicted from the cache.

LRU: The item that has been unused for the longest period of time is replaced.

LFU: The item that has been accessed least frequently is removed from the cache.

Types of Caching

in-memory caching, database caching, web-server caching and client-side caching.

Spring base caching. Annotations used:

1. @CacheConfig provides a mechanism for sharing common cache-related settings at the class level.
2. @EnableCaching enables Spring's annotation-driven cache management capability.
3. @Cacheable
4. @CacheEvict annotation to remove values so that fresh values can be loaded into the cache again.
5. @CachePut annotation can update the content of the cache without interfering with the method execution.
6. @Caching is used in the case we want to use multiple annotations of the same type on the same method.

Guava Cache is used for application-level caching.

For the first time, it took time to fetch the records, as they had to be fetched from the backend. Also, the data was written into the cache. Next time onwards, the data was fetched from the cache itself.