

## Redis → Expiry Strategies

### Ejection Strategies:

↳ Redis stores data in memory (RAM), which is limited

↳ When Redis reaches 100% RAM usage, it must "evict (remove)" some keys to make space.

### # Strategies:

#### ① TTL (Time-to-Live)

↳ a key is deleted automatically when its expiration time is reached

Ex: In the web sessions

#### ② LRU (Least Recently Used)

↳ evicts the key that was not accessed for the longest time

Ex:

e-commerce app with millions of product pages

Hot products = accessed frequently

Old products = few views

\* Redis keeps frequently accessed products in memory and removes the old ones.

Imp: LRU prevents removing a frequently used item - high cache hit rate

#### ③ LFU (Least Frequently Used)

↳ evicts the key accessed the least number of times, regardless of recency.

Ex: Trending / popular content

viral feeds

leaderboards

ads ranking

#### ④ FIFO (First in First Out)

↳ evicts the key that was inserted first, regardless of usage.

Ex:

batched data storage

→ only need recent logs;

older logs can disappear.

#### ⑤ Random Eviction

↳ choose any key to delete

- no logic

use-case: High throughput + non-critical data

Ex:

batching analytics counters

IOT sensor data ingestion

Temporary ML model features

#### Advantages:

extremely fast - no update check

good for heavy-write systems