Search Index vs Hydration vs Feature Store Cheat Sheet

Quick reference for designing efficient retrieval, ranking, and serving layers

1. Core Principle

Put in the index only what you need to find, filter, rank, or render minimally.
Everything else belongs to hydration or the feature store.

2. Quick Comparison

Aspect	Search Index	Hydration (DB/Cache)	Feature Store
Purpose	Fast retrieval, filtering, first-pass ranking	Full data enrichment and display	Online feature access for ML ranking and shared consumers
Typical Data	• IDs, titles, keywords • Facets, filters • Coarse numeric features • Static ranking features • Small display info	• Full description • Dynamic fields (price, stock) • Large JSON or media • PII or sensitive data	• CTR, CVR, engagement • Short-term aggregates • Cross-entity signals • Model features
Update Frequency	Low to moderate (hours or daily)	High (seconds to minutes)	High to very high (real-time)
Size Limit	Small (compact doc)	Large objects ok	Moderate (feature vectors)
Latency Need	Milliseconds (query time)	Only for top-K hydration	Microseconds to ms (online inference)
Consistency Goal	Eventually consistent, stale tolerant	Strong source of truth	Fresh, low-latency consistency
Consumers	Search retrieval service	API / frontend rendering	ML ranking, recsys, ads

3. Field Placement Rule of Thumb

- **In Index:** essential for recall, filter, sort, or first ranking.
- **In Hydration:** large, volatile, or rarely accessed.
- **In Feature Store:** frequently updated metrics or shared model features.

Ask for each field:

1. Needed to select or rank? $\rightarrow$ Index
2. Updated often? $\rightarrow$ Avoid index
3. Large? $\rightarrow$ Hydration
4. Shared or real-time? $\rightarrow$ Feature store

4. Serving Flow Pattern

1. **Candidate Retrieval** (Index) Fetch top-N doc IDs using textual/semantic matching and indexed filters.
2. **Hydration** (DB/Cache) Fetch detailed data for top-N from fast key-value or DB.
3. **Feature Enrichment** (Feature Store) Join real-time model features for top-M.
4. **Final Ranking** Run ML model with combined features and return results.

5. Example Split for E-commerce

Index	Hydration	Feature Store
ID, title, short name, category, brand, price bucket, rating bucket, availability flag, popularity score	Full description, high-res images, reviews, seller details, exact inventory count, last-updated timestamps	CTR/CVR, trending score, user-item interaction features, recent engagement counters

6. Notes

- Keep the index small and stable—optimize for recall and speed.
- Avoid frequent reindexing of volatile data.
- Use a cache or feature store for freshness and reusability.
- Design coarse fields (for example, price_bucket, in_stock) to avoid heavy updates.
- Ensure all paths (index, hydration, features) share consistent identifiers.

"Retrieve with index, enrich with data, rank with features."