SCORED.TOOLS · ROUNDUP

Best AI Database Platforms 2026: Vector & Hybrid Search Ranked

We tested the leading AI databases on ingestion speed, hybrid search quality, scaling, and price. Here's how Pinecone, Weaviate, Qdrant, Chroma, Milvus, pgvector, LanceDB, and Turbopuffer actually compare.

If you're building anything that retrieves context for an LLM — RAG, agent memory, semantic search, recommendation — your database is the part that decides whether the system feels fast and accurate or slow and dumb. Model choice gets the attention. The vector store does the work.

We spent the past few weeks running the same workload (10M embeddings, 768-dim, mixed metadata filtering, hybrid BM25 + dense queries) against the eight platforms below. Rankings reflect latency at p95, recall at the operating point we'd actually ship, and what it costs to run the thing for a year. No vendor here paid for placement, and a couple of the lower-ranked ones are products we use ourselves — the ranking is the workload, not the loyalty.

1. Pinecone — Score: 9.2/10

Pinecone is still the default "it just works" managed vector DB, and the serverless tier launched in 2024 finally fixed the pricing model that used to make it painful at small scale. Ingestion is fast, query latency is consistently sub-50ms p95 on our test set, and the new sparse-dense hybrid index ships out of the box instead of requiring you to wire two stores together. The only real friction is that you're locked into their infrastructure — no self-host story, and the metadata filter language is more limited than what you get from a SQL-based option.

Best for: Teams that want a production-grade vector DB on day one and don't want to think about ops.

Pricing: Serverless from $0 with usage-based billing (~$0.33/M reads, $4/M writes, $0.33/GB-month storage). A typical 10M-vector RAG workload lands around $70–150/month.

2. Turbopuffer — Score: 9.0/10

The newcomer that's actually earned the hype. Turbopuffer is built on object storage (S3) instead of attached SSDs, which sounds like it should be slow but isn't — they've engineered around the cold-

read problem well enough that p95 is competitive with Pinecone at a fraction of the cost. The pricing is genuinely disruptive: storage at object-storage rates, and you only pay for queries you actually run. The trade-off is feature surface — fewer index types, no GraphQL or fancy hybrid orchestration, just a clean API that does exactly one thing well.

Best for: Cost-sensitive workloads with bursty or read-heavy access patterns, especially at large corpus sizes.

Pricing: ~$0.04/GB-month storage, $0.10/M queries, $1/M writes. Same 10M-vector workload runs around $15–40/month.

3. Qdrant — Score: 8.7/10

Qdrant is the right answer when you want self-hosted control without writing your own retrieval layer. The Rust core is fast, the filter pushdown is the best in the category (complex metadata predicates don't fall off a cliff like they do in some competitors), and the managed Cloud option exists if you decide later that ops aren't worth your time. Quantization options — scalar, product, binary — let you trade recall for memory in a much more granular way than other platforms allow.

Best for: Teams that need self-hosting (compliance, data residency, cost) or want fine-grained control over the retrieval pipeline.

Pricing: Free self-hosted (open source). Cloud from $25/month for a starter cluster; scales linearly with RAM.

4. Weaviate — Score: 8.5/10

Weaviate is the most batteries-included option here. Built-in modules for embedding generation, hybrid search with BM25, multi-tenancy as a first-class concept, and a generative-search feature that calls an LLM with retrieved context in a single API call. It's a lot — sometimes more than you need — but for teams building multi-tenant SaaS with RAG features, the multi-tenancy model alone is worth the price of admission. Query latency is slightly behind Pinecone and Qdrant under heavy filter load, but the feature surface compensates.

Best for: Multi-tenant applications and teams that want hybrid search and generative integration without assembling it themselves.

Pricing: Open source self-hosted. Serverless Cloud from $25/month; Enterprise Cloud from ~$2,500/month.

5. Pgvector — Score: 8.3/10

If you already run Postgres, Pgvector is almost always the right starting point. The HNSW index added in 0.5 closed most of the performance gap with dedicated vector DBs, and being able to join vector search results against your transactional tables in a single query is a capability the purpose-built stores simply can't match. It falls behind at the extremes — billions of vectors, sub-10ms latency requirements — but for the 90% of workloads that don't need those, adding another database is just operational debt you don't have to take on.

Best for: Anyone already on Postgres. Especially good when vector results need to be combined with relational data.

Pricing: Free extension. Cost is whatever you already pay for Postgres (Supabase, Neon, RDS, self-hosted).

6. Milvus — Score: 8.1/10

Milvus is the heavyweight option — designed from the start for billion-vector workloads with a disaggregated architecture (separate compute, storage, and coordinator nodes). At small scale it's overkill and the operational complexity will hurt you. At very large scale it's one of the only options that holds up. The Zilliz Cloud managed version smooths most of the rough edges, but you'll still feel the architectural weight in setup and tuning.

Best for: Workloads with 100M+ vectors where horizontal scaling matters more than time-to-first-query.

Pricing: Free open source. Zilliz Cloud serverless from $0 with usage billing; dedicated clusters from ~$99/month.

7. Lancedb — Score: 7.8/10

Lancedb is the embedded-first vector DB — runs in-process like SQLite, stores data in the Lance columnar format on disk or in object storage, and scales from a single-file local dev setup to a distributed deployment without changing your code. The developer experience is excellent for Python and TypeScript workflows, and the multimodal support (vectors + images + raw bytes in the same table) is genuinely useful. The managed cloud product is still maturing, so for production at scale you're either self-hosting or waiting.

Best for: Local-first development, multimodal data, and applications that want to ship a vector DB inside the application binary.

Pricing: Free open source. LanceDB Cloud in preview with usage-based pricing.

8. Chroma — Score: 7.5/10

Chroma won the prototype-and-demo phase by being the easiest possible thing to install and use — three lines of Python and you have a working vector store. That's still true, and for the first 100K vectors of a project you should probably just use it. The recent Chroma Cloud release is a credible production story, but the platform is still catching up on the operational features (sharding, advanced filtering, multi-tenancy) that the older platforms have refined over years.

Best for: Prototypes, notebooks, RAG demos, and small production workloads where simplicity wins over scale.

Pricing: Free open source. Chroma Cloud from $0 with usage-based pricing (~$0.40/GB-month, $2.50/M writes).

Comparison Table

Platform	Score	Self-host	Hybrid Search	Best Scale	Starting Price
Pinecone	9.2	No	Yes	1M – 1B	$0 serverless
Turbopuffer	9.0	No	Limited	10M – 10B	$0 usage-based
Qdrant	8.7	Yes	Yes	100K – 1B	Free / $25 cloud
Weaviate	8.5	Yes	Yes (best-in-class)	1M – 500M	Free / $25 cloud
pgvector	8.3	Yes	Via FTS	10K – 50M	Free (your Postgres)
Milvus	8.1	Yes	Yes	100M – 10B+	Free / $99 cloud
LanceDB	7.8	Yes (embedded)	Yes	1K – 100M	Free open source
Chroma	7.5	Yes	Limited	1K – 10M	Free / usage cloud

Final Picks

If you want the fastest path to production: Pinecone. Sign up, ingest, ship. You'll spend more than you would self-hosting, but you'll spend zero engineering hours on the database.

If you're cost-sensitive at scale: Turbopuffer. The economics on object storage genuinely change what's affordable, and the team is shipping fast enough that the feature gap is closing every month.

If you're already on Postgres: Pgvector. Don't add another database until you've measured that pgvector can't handle your workload. For most teams, that day never comes.

If you need self-hosted control: Qdrant. The filter performance, quantization options, and operational simplicity make it the cleanest self-host story in the category.

If you're building multi-tenant SaaS: Weaviate. The multi-tenancy model is the right primitive and you'll feel its absence anywhere else.

The thing nobody tells you when you're picking a vector database: the best one is the one that lets you iterate on the rest of your retrieval pipeline — chunking, embedding choice, reranking, prompt design — without becoming the bottleneck. All eight of these will hold up at the scale most teams actually operate at. Pick the one whose operational model fits your team and move on to the problems that actually matter.
