

Open Table Format Decision Guide

Apache Iceberg vs Delta Lake vs Hudi

Vertex Frontier
Decision Workbook Companion
2026-08-19

A practical way to choose a table format without confusing format capability, engine behavior, catalog ownership, and platform convenience.

Contents

1	How to use this guide	1
2	One-page decision filter	1
3	The five-layer model	1
4	Workload decision matrix	2
5	Format scorecard	2
6	Compatibility matrix	3
7	Fair benchmark protocol	3
7.1	Control variables	4
7.2	Workload families	4
7.3	Benchmark result template	4
8	Maintenance ownership scorecard	4
9	Migration and exit path	5
10	When none of the three is right	5
11	Official source checklist	6

1 How to use this guide

This guide is the printable companion to the Vertex Frontier comparison of Apache Iceberg, Delta Lake, and Apache Hudi. It is intentionally a decision tool rather than another feature checklist. Use it when a platform team needs to shortlist a format, plan a benchmark, review maintenance ownership, or document an exit path.

The workbook has one central rule: a format is not the whole system. Record the table format, physical files, catalog, execution engines, managed platform, and operators as separate decisions. A table may be readable by an engine while updates, deletes, time travel, maintenance, or recovery remain unavailable or unowned.

Evidence boundary. This guide separates documented capabilities from vendor positioning, editorial analysis, and claims that remain unverified until tested in your exact engine and version. It contains no fabricated benchmark results.

2 One-page decision filter

Use this first table to identify which format deserves the earliest test. It is a shortlist, not a final verdict.

Dominant requirement	Start with	Validate before committing
Open, multi-engine analytics	Iceberg	Catalog operations, DDL/DML, time travel, maintenance, and permissions in each required engine.
Databricks- or Spark-first platform	Delta Lake	Open-format behavior versus platform-specific features, readers outside the platform, and the exit path.
High-rate CDC and keyed upserts	Hudi	Record keys, indexing, incremental reads, compaction, clustering, cleaning, and reader cost.
No dominant constraint	Controlled bake-off	The benchmark protocol, total operating cost, and the team that will own maintenance.

Decision rule: If you cannot name the workload, writer, readers, catalog owner, and maintenance owner, you are still choosing a platform story—not a table format.

3 The five-layer model

Use the following model whenever a comparison starts to blur format features with platform features.

Layer	What it controls	Examples and decision risk
Storage files	Rows, encoding, compression, and file statistics	Parquet and ORC. Files alone do not provide table commits or durable snapshot history.

Table format	Snapshots, commits, schema and partition metadata, change semantics, and maintenance model	Iceberg, Delta Lake, Hudi. Engine exposure may be partial.
Catalog	Identity, registration, namespaces, permissions, and metadata location	REST Catalog, Hive Metastore, Glue, Unity Catalog, Nessie.
Execution engine	SQL, scans, writes, merges, deletes, joins, planning, and maintenance	Spark, Trino, Flink, Athena, Snowflake.
Managed platform	Automation, optimization, governance, observability, and extensions	Databricks and cloud platforms may add value that is not a format property.

4 Workload decision matrix

Score the workload before scoring the format. For every answer, record the normal week and the bad week.

Question	Iceberg signal	Delta signal	Hudi signal
Do several engines need to operate the same table?	High value when portability and partition evolution matter.	Validate reader and DML consistency outside the core platform.	Validate reader behavior and MOR/COW semantics across engines.
How much CDC or keyed upsert activity exists?	Test delete representation and compaction cost.	Test merge cost, checkpointing, and platform optimization.	Test record keys, indexes, incremental reads, and services.
Will the partition strategy change?	Partition evolution is a strong reason to test Iceberg.	Check platform partitioning and clustering behavior.	Check partition layout and clustering impact on old data.
Is the dataset small and append-only?	A table format may still help if governance or time travel matters.	Existing platform convenience may be unnecessary overhead.	Built-in mutation services may exceed the workload need.

5 Format scorecard

Rate each format from 1 to 5 for your environment. A 5 means “strong fit supported by our evidence,” not “best in the market.”

Criterion	Weight	Iceberg	Delta	Hudi	Evidence note
Multi-engine portability	15				
CDC and keyed upserts	15				
Spark / Databricks fit	10				
Schema and partition evolution	10				

Streaming and incremental reads	10				
Catalog portability	10				
Maintenance ownership fit	10				
Governance and exit path	10				
Small-file and indexing needs	5				
Team familiarity	5				

Weighted result = $\text{sum}(\text{weight} \times \text{rating}) \div \text{sum}(\text{weights for completed ratings})$. Leave a cell blank when you do not have evidence; do not enter zero to represent “unknown.”

6 Compatibility matrix

The word “supported” is too vague for production. Use three states: **readable**, **operationally supported**, and **owned by the platform team**.

Operation	Iceberg	Delta Lake	Hudi	Record in your test log
Read current snapshot	Status / engine / catalog	Status / engine / connector	Status / table type	Query and result
Append	Status / conflict behavior	Status / log behavior	Status / keying	Throughput and files per commit
MERGE / upsert	Status / delete representation	Status / merge implementation	Status / record keys	Replay correctness and cost
Delete	Status / reader behavior	Status / operation and retention	Status / hard or soft delete	Latest-state correctness
Time travel / rollback	Snapshot retention	Version history	Timeline / instant	Recovery time and permissions
Maintenance	Snapshots, orphan files, rewrites	Cleanup and optimization	Compaction, clustering, cleaning	Owner, trigger, failure action

Dated baseline reminder. Compatibility changes faster than comparison articles. Add a review date and the exact format, connector, catalog, and engine versions to every completed matrix.

7 Fair benchmark protocol

A benchmark should answer a decision question, not produce a winner’s badge. If a result cannot change an architecture decision, it does not belong in the benchmark.

7.1 Control variables

Keep the logical dataset, object-store class, compression assumptions, partition intent, record distribution, compute shape, engine version, concurrency, cache state, and maintenance policy visible. If a format requires a different operating model, record that difference rather than hiding it.

7.2 Workload families

Test	What to run	Primary measures
B-01 Append scan	Append data, then run representative scans.	Planning latency, p95 query latency, files created, cost.
B-02 CDC replay	Replay inserts, updates, deletes, duplicates, and late events.	Ingest-to-query lag, replay correctness, read amplification, pending changes.
B-03 Backfill + recovery	Rewrite older partitions and interrupt a commit or maintenance job.	Recovery time, failed retries, data correctness, maintenance backlog.
B-04 Multi-engine read	Run the same acceptance queries in every required reader.	Result parity, planning time, permissions, unsupported operations.

7.3 Benchmark result template

ID	Format + version	Engine + version	Operation mix	Primary metric	Result	Cost	Maintenance state
B-01							
B-02							
B-03							
B-04							

Do not record a format as “faster” without naming workload, versions, hardware, row shape, concurrency, cache conditions, maintenance state, metric definition, and raw result.

8 Maintenance ownership scorecard

Every format creates work. The differences are who owns it, how visible it is, and what happens when it fails.

Control	Owner	Trigger	Failure action
Snapshot or version retention		Age or storage threshold	Pause cleanup; verify recovery window.
Orphan-file cleanup		Age beyond maximum write duration	Stop deletion and reconcile metadata.
Compaction or rewrite		Small-file or read-amplification threshold	Throttle, retry, or quarantine the partition.

Clustering or indexing		Query or write threshold	Record backlog and protect ingestion.
Metadata cleanup		Metadata growth threshold	Preserve a recovery copy before deletion.
Catalog health		Failed load or permission alert	Escalate before changing table state.

9 Migration and exit path

A good format decision includes a way to discover that the decision was wrong.

Phase	Evidence to collect	State	Owner / notes
Inventory	Schemas, keys, partitions, files, catalog entries, permissions	Go / Hold	
Parallel read	Query parity, null rates, aggregates, time travel	Go / Hold	
Shadow or dual write	Commit behavior, lag, duplicates, retry and recovery	Go / Hold	
Cutover	Freeze or coexistence plan, rollback pointer, communications	Go / Hold	
Post-cutover	Freshness, error rate, maintenance backlog, cost	Go / Hold	
Exit / rollback	Known-good version, catalog path, data ownership, trigger	Rollback	

Exit-path questions: Can another engine read the table? Can the catalog change without rewriting data? Can the team restore a known-good snapshot or version? Which metadata or operations are format-specific? Who owns the final decision to migrate back?

10 When none of the three is right

An open table format is not a universal replacement for OLTP, high-concurrency point updates, strict cross-table transactions, or sub-second indexed serving. A relational database, key-value store, search system, or serving warehouse may be a better layer for those requirements.

A simple Parquet layout can also be reasonable for a small, append-only dataset written by one trusted job when the data is cheap to replay and the team does not need table-level commits, time travel, schema enforcement, concurrent writers, or managed governance. The correct decision is the least complex system that still meets correctness and operating requirements.

11 Official source checklist

Verify date-sensitive claims against the source pages before publishing or refreshing the workbook:

- [Apache Iceberg releases](#)
- [Apache Iceberg maintenance](#)
- [Delta Lake documentation](#)
- [Delta Lake porting guide](#)
- [Apache Hudi releases](#)
- [Apache Hudi Spark support matrix](#)

Vertex Frontier · Open Table Format Decision & Architecture Workbook