

# Endatabas Executive Summary

**Problem: Updates are destructive** – Most OLTP databases today are *mutable*. They destroy data on every `UPDATE` or `DELETE` statement executed by users and applications. This made sense in the 1980s when disk space was expensive — but in the 2020s, it makes compliance and audits difficult and expensive. It makes time-travel impossible in the database. Instead, it is an expensive chore for developers of the application.

**Solution: Full History** – Endatabas is an *immutable* database. Facts aren't lost when new facts appear; everything is retained. Yet GDPR-compliance is as simple as a single `ERASE` statement. Audits and time travel are free and effortless.

**Problem: SQL databases are expensive to change** – Most SQL databases contain no timeline and lack flexible schema, causing them to calcify with age. The alternative is JSONB columns, which make the database fragile and unsafe. It can be impossible to build on top of old SQL databases, leading to expensive workarounds.

**Problem: Document databases are non-standard** – During the 2010s, NoSQL and document DBs had their moment in the sun. But they all use proprietary query languages or a manual SQL mapping layer. They are unsafe due to the storage of weakly-typed JSON documents. Developers can't reuse their knowledge of SQL, leading to training expenses. Vendor lock-in prevents customers from moving elsewhere.

**Solution: Documents in SQL** – Endatabas is a SQL database with a relational storage/query model. However, it also supports native, strongly-typed documents. These documents are stored in Apache Arrow format and queried with standard SQL.

**Problem: Fragmented and expensive analytics** – For all but the simplest cases, analytics happen in separate tools like time-series and columnar databases. Duplicating data from OLTP databases and event streams into these tools adds complexity and expense. Newer ideas like Hybrid Transactional/Analytical Processing (HTAP) only add column stores to row-oriented OLTP stores or, worse, force the user to choose between them. They don't solve the expensive data duplication problem.

**Solution: True HTAP** – Endatabas unifies OLTP, timeline, and columnar data into one relational model. No copying data. No additional tools.

**Problem: Storage is coupled to compute** – All but the newest databases couple storage to compute nodes. Local disk is limited and expensive and horizontal scaling can only be achieved with sharding. High Availability, automatic fail-over, and horizontal scaling are all complicated and expensive under this model.

**Solution: Separate Storage from Compute** – The Endatabas Object Store is the ultimate source of truth. Normal database nodes are cheap and scale freely, both up and down. This technology is sometimes referred to as a “serverless” database.

**Problem: Operations are expensive** – Developers, data scientists, data engineers, and DBAs are increasingly forced to babysit databases. Database users spend more and more time optimizing queries and fiddling with `CREATE INDEX`. Cloud infrastructure makes it easy to create databases — but much more expensive to run them.

**Solution: Light and Adaptive Indexes** – Machine Learning for indexes is now possible. Rather than forcing users to tune the database themselves, the database learns its own usage patterns — across storage and compute — and, in turn, automates operations.

## Why now?

The combination of these technologies wasn't possible ten years ago. With the advent of contemporary research and commercial databases demonstrating some of these solutions, it has only recently become clear that such a database could even be built. This database can't be built with Postgres plugins. It requires a total reexamination.

## Why Endatabas?

All of the solutions presented above stand together in a single, cohesive whole. For example, immutable data calls into question the idea of a single schema for a table's entire lifetime — how can users examine data from 4 years ago, exactly as it was back then? This requires dynamic schema, and a dynamic SQL dialect to support it. So it is with all of the solutions: immutable documents, columnar storage, separated storage and compute, and light and adaptive indexes all support true HTAP in a way no other database can. Only Endatabas solves all these problems with a single tool.