Endatabas Investment Memo

Endatabas is a SQL document database with full history. We will run Endatabas (sometimes abbreviated *Endb*) as a fully managed service for our customers.

There is an aphorism: "never write your own database." In response, we would argue that there are certain capabilities we can only provide if we *are* the database — and that the time is right for Endatabas. The remainder of this memo will explain why.

Basics

- Stage: Pre-Seed / Seed
- Target: $2-4m USD
- Capital: Self-funded
- Team Size: 2
- Location: Sweden and Canada
- Website: https://www.endatabas.com
- Live Demo: https://www.endatabas.com/demo
- Source Code: https://github.com/endatabas/endb (AGPL-3.0)
- Email: hello@endatabas.com

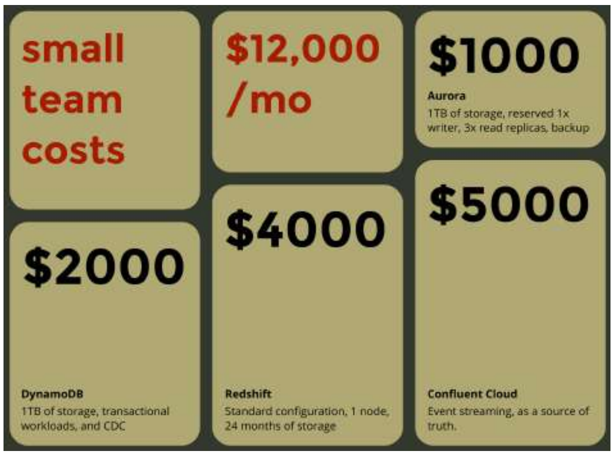
Problem

The data product market is huge, complicated, and expensive. It has created a landscape of splintered and scattered data tools, compounding costs, and wasted employee time.

There are 50 logos on the right. You probably recognize most of them. Across the industry, a "wave of consolidation" is predicted for the data product market. [i] We want to stand where that wave of consolidation will land.

Customer Pain

Here are some sample figures for a small team with modest cloud data needs. These are the costs which force consolidation.


small team costs

$12,000 /mo

$1000
Aurora
1TB of storage, reserved 1x writer, 3x read replicas, backup

$2000
DynamoDB
1TB of storage, transactional workloads, and CDC

$4000
Redshift
Standard configuration, 1 node, 24 months of storage

$5000
Confluent Cloud
Event streaming, as a source of truth.

None of these figures even account for maintenance costs. Companies are bleeding money to (a) tune the database and (b) optimize queries in their apps. To reduce cloud costs, they have to spend more money on staff. This is a data product addiction.

Databases (and this addiction) are everywhere — every industry and government — and cloud databases represent well over 90% of database market growth in 2023. The cloud has democratized who can set up a database. However, this sets in motion a cycle of people buying data tools, which need tuning, which requires more people, and so on. As Håkan says: “if you want the database to be tuned at all, it needs to tune itself.”

Not only is the market splintered, but database products are also splintered, internally. SQL databases store documents as weakly-typed JSONB columns. Document databases provide pseudo-SQL with mapping layers. Vector databases, which are more a feature than a product, have created an entirely new sub-category of document databases. And while many incumbent databases now provide vector embeddings, the problem of the SQL / document divide remains.

Many companies we speak to feel their data architecture is like a five-car garage. Even if that five-car garage didn’t feel so wasteful back in 2020, they all know they need to downsize now. But most don’t know where to start.

Truth be told, many of the 50 logos on page one are akin to car parts manufacturers more than car manufacturers: they are easy costs to eliminate with a more reliable car.

Solution

In Swedish, *endatabas* means both *a database* and *one database*.

To understand what we mean by “one database”, let’s explode the architecture of the Endatabas tagline (“SQL document database with full history”) into its components.

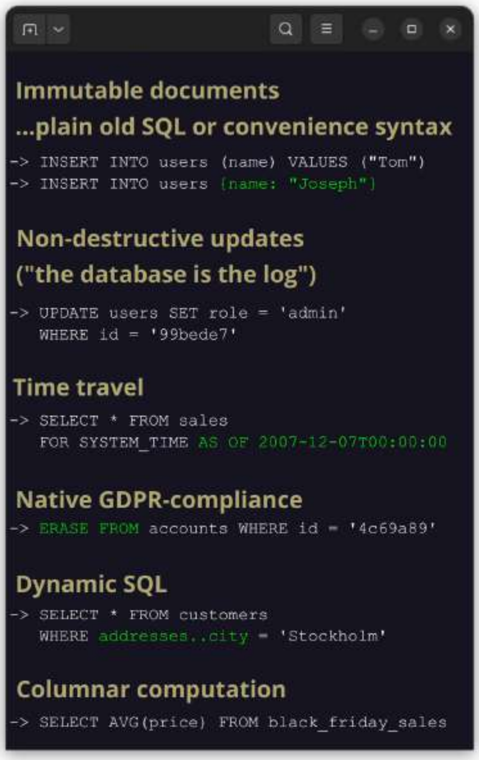
SQL Document Database means dynamic columnar storage, separated storage and compute, and adaptive indexes. **Full History** entails immutable data, time travel queries, and built-in GDPR-compliance. That may sound like a lot but it’s actually slicing through layers of complexity by getting to the root of the problem. These are the *pillars* of Endb, and they stand together in one coherent whole.

It might also appear that Endb is multi-model. It’s not. Endb unifies the document, columnar, and temporal models into the relational model.

In 2004, Michael Stonebraker wrote, in *What Goes Around Comes Around*, that new models and niche databases repeatedly emerge, then consolidate into the relational model, roughly every 10 years. [ii] Andy Pavlo is preparing a follow-up paper (*What Goes Around Comes Around And Around*) which confirms Stonebraker’s thesis. Stephen O’Grady agreed in his 2021 Redmonk article, *A Return to the General-Purpose Database* [iii] and Martin Kleppmann said in an interview during GOTO Amsterdam 2023 that he felt relational and document databases were converging. [iv] Contemporary opinions reflect Stonebraker’s original thesis: the relational model will always be king.

UX / DX

Almost every pillar of Endb is exposed to SQL — users have power normally only granted to Ops.



Immutable documents
...plain old SQL or convenience syntax

```
-> INSERT INTO users (name) VALUES ("Tom")
-> INSERT INTO users {name: "Joseph"}
```

Non-destructive updates
("the database is the log")

```
-> UPDATE users SET role = 'admin'
   WHERE id = '99bede7'
```

Time travel

```
-> SELECT * FROM sales
   FOR SYSTEM_TIME AS OF 2007-12-07T00:00:00
```

Native GDPR-compliance

```
-> ERASE FROM accounts WHERE id = '4c69a89'
```

Dynamic SQL

```
-> SELECT * FROM customers
   WHERE addresses..city = 'Stockholm'
```

Columnar computation

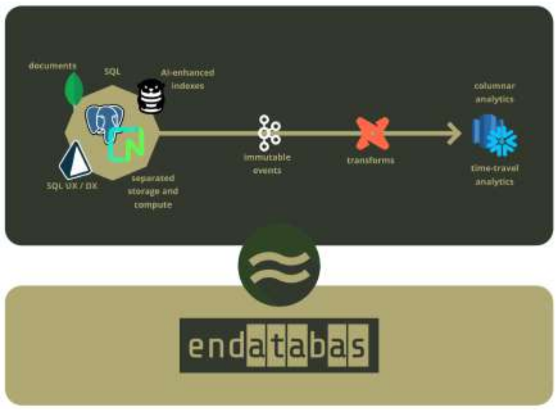
```
-> SELECT AVG(price) FROM black_friday_sales
```

Under Endb's immutable documents, the classic "the log is the database" — popularized by the 2017 Aurora paper [v] — is flipped on its head: the database is the log. This gives time travel to our users, but it also implies the existence of an `ERASE` statement for GDPR compliance.

Dynamic SQL affords users a lot of power. In this recursive query, we've added a light document-native syntax within the relational model.
Columnar computation happens for free and requires no special syntax.

Endb's SQL dialect is built on top of SQLite's, to keep our SQL both light and familiar. Small conveniences (non-standard SQL) are highlighted in green in these examples.

Product Analogy



Here are some example products Endb can eliminate, replace, or augment. Endb has a small SQL footprint to replace larger SQL grammars and eliminates SQL mapping. Endb solves as many data problems with one product as possible by providing SQL for: documents, event streams, transforms, and columnar analytics. Last, Endb automates Ops, index tuning, and time travel.

Endb doesn't need to replace all these tool archetypes for every use case. If it can replace one or two, its value is self-evident.

Beta and Benchmarks

After 14 months of development, the Endb core is fully-functional and released in Beta. You can easily try it yourself to get a feel for the dynamic SQL seen on the previous page:

https://www.endatabas.com/demo

Endb currently passes SQL Logic Test and TPC-H. The TPC-C benchmark is also running against Endb, so we can expand our concurrency models within it.

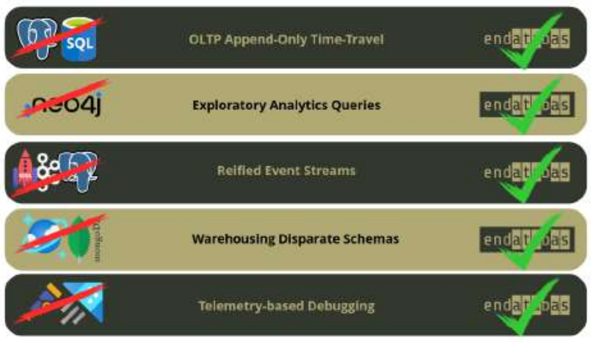
We score 81% on the SQL Acid Test. (For context, only Postgres and Umbra achieve 100%. Oracle scores 41%, SQL Server scores 26%, and MySQL scores 15%.) Our remaining 19% is well-known to us, as these test failures stem from explicit dynamic SQL choices in Endb.

Solution: One Database

Our aim is to produce very high utility software. We want to be the Post-Postgres: quiet, resilient, and chosen by default.

Tuner cars and helicopters will continue to exist, but we want to serve the 80% of problems in the middle.

Example Go-To-Market Use Cases


SQL
OLTP Append-Only Time-Travel
endatabas
neo4j
Exploratory Analytics Queries
endatabas
Reified Event Streams
endatabas
Warehousing Disparate Schemas
endatabas
Telemetry-based Debugging
endatabas

Of course, Postgres isn’t going to disappear tomorrow and we won’t built true HTAP in our first year. We need to find niche use cases where our early customers will get value out of Endb immediately.

In each of the following cases, we're replacing non-standard or incomplete query solutions built on top of disparate storage formats by developers forced to create custom solutions to common problems. We replace these with standard SQL over HTTP storing Apache Arrow. Each of these use cases is *automatically* solved by Endb.

OLTP Append-Only Time-Travel

Almost every software team creates append-only tables in traditional SQL databases. This pattern leads to awkward, bespoke, and complicated time-travel queries.

Exploratory Analytics Queries

We've spoken to many teams who use dynamic stores like Neo4j for exploratory queries over arbitrary data. But this data almost always begins its life as JSON or XML and NoSQL databases are either restricted by non-standard data models (like Neo4j) or inflexible and non-standard query languages (like MongoDB).

Reified Event Streams

By far the most common way teams store historical data today is with event streams like Kafka. But event streams either provide no query functionality at all or they add query functionality with something like KSQL or bespoke append-only tables, which do not support SQL:2011 or standard time-travel queries.

Warehousing Disparate Schema

Almost every company we know has a MongoDB installation for oddball data: mismatched schemas, schemaless, and/or semi-structured documents. But no one is happy with it. And it suffers from a lack of standardized query language.

Telemetry-Based Debugging

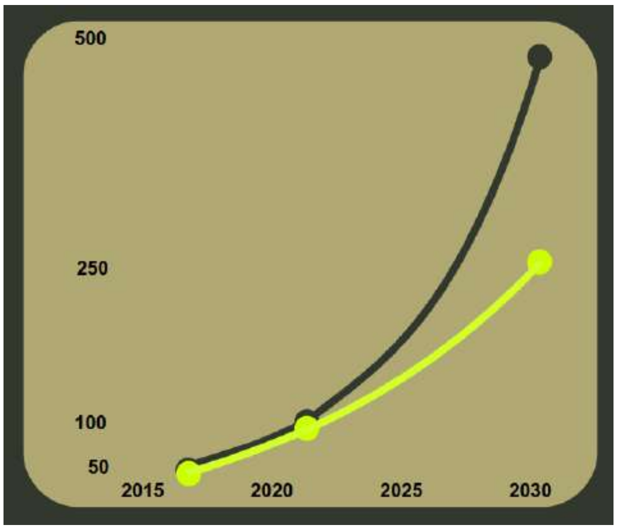
SaaS and hybrid products require that development and support teams can diagnose issues when they arise. Without the complete history of an application's state and behavior, developers must resort to hand-rolling audits. This telemetry data usually only lives within retention windows, limiting its usefulness.

Market

At a glance, the total addressable DBMS market was $80 billion USD in 2022. And it is still growing.

Even though businesses are trying to consolidate their data products and associated spend, data is continuously created and the pie keeps growing. By some estimates, the combined cloud storage and DBMS markets will approach a trillion dollars in the 2030s. Of course, these are giant markets — but by 2030, we hope to participate in them.

The DBMS market (bright green line) was estimated by Gartner at $38 billion in 2017, with a CAGR of 16%. [vi] As of 2022, it was estimated at $80 billion, which Gartner described as "unprecedented growth." [vii] The projected 2030 DBMS market is $262 billion.



The cloud storage market (dark green line) is even more lucrative with Fortune Business Insights estimating the 2030 market at $472 billion based on 23.4% CAGR. [viii]

Market Opportunity Analysis

After speaking to dozens of companies about their data problems, it is clear the opportunity to capitalize on consolidation is very real. Customers want a simpler product, to replace NoSQL stores with a SQL-native store, and to integrate legacy schemas into one store.

We put everything in one place: immutable data in-the-large, infinite storage, adaptive indexing, and hybrid transactional/analytical processing (HTAP). The time is right.

Growth

We will start with small customers (on the order of $10,000/mo) who want to consolidate products and enable small teams with dynamic SQL. We frequently hear that schema changes are a problem for small teams in large organizations (banks, healthcare, governments) where such changes require multiple people or teams.

We can grow by reducing the HR burden. Starting small as a convenient SQL-native document store, we will work to become the default data solution through immutable data and analytics. Medium sized customers (OTOO $50,000/mo) want HTAP and adaptive indexes.

Eventually, we will leverage our ground-up build to compete with large players like Oracle and AWS by 2030.

Positioning

We sell simplicity. One Database means users (developers, data scientists, data engineers, etc.) stay in — and on — the product longer, while we visibly reduce costs. SaaS helps us grow the product in new ways: the more data we have, the more AI models for adaptive indexes benefit.

We sell the dream of Aurora, but actually help customers manage costs instead of becoming a new burden.

Competitors

As a jack-of-all-trades database, Endb competes with many companies. Our competitors include Aurora and Neon (Postgres), MongoDB, Neo4j, Cosmos DB, InfluxDB, Materialize, Firebase, Fauna, Snowflake, and SingleStore, to name a few.

Our direct competitors do not include data products with narrower USPs like Spanner, Cockroach, ClickHouse, Elastic, VoltDB, or dbt.

Defensibility

Aurora and Neon are built on Postgres. While this is a solid foundation, building on Postgres is tactical — it's extremely difficult to grow or evolve. We are not just another Postgres vendor.

Everyone we speak to wants to keep their document stores but they are all tired of NoSQL. MongoDB, Neo4j, and Cosmos DB all have ad-hoc query languages or a fragile SQL translation layer. Endb is the only SQL-native document database.

Immutable by default is possible in 2024, but almost only found in event sourcing so far. It can (and should) happen in the database. InfluxDB and Materialize are secondary tools. Endb keeps immutable data and the timeline in the primary data store.

Customers are tired of building on proprietary APIs and proprietary tools. We provide the convenience of Firebase and Fauna with a standards-based tool. Endb is open source and speaks standard SQL over HTTP and WebSockets.

Against Snowflake and SingleStore, open source is noteworthy. IP is different in 2024. The HashiCorp / TerraForm fiasco last year demonstrated that consumers want true open source infrastructure products, not lip service. As a result, ownership (copyright) and restrictive licenses matter now more than ever. Endb starts with a clean slate.

Why Endatabas Wins

Endb is licensed AGPL-3.0, which IP lawyer Martin Clausen refers to as "Ultra Restrictive Copyleft." [ix] Additionally, our tech and market research has as much to do with distilling the problem space as it has to do with the number of papers we've read or interviews we've conducted. Even so, years of research can't be reproduced overnight. If companies understood the opening in the market, Endb would exist already.

Beyond hard and soft IP, incumbents are calcified. SQL databases struggle to build dynamic SQL. Document databases can only add band-aids. For our competitors, building HTAP means cobbling column storage onto legacy OLTP stores.

Pricing

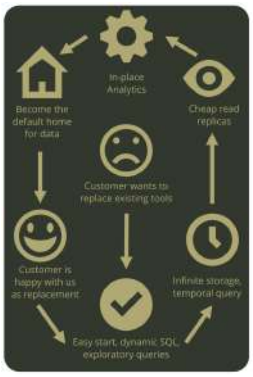
Endatabas will be SaaS-first, for which we will have standard unit-measured, usage-based pricing which bundles compute and storage. This makes it easier for customers to start and grow their usage.

On-premises is also an option, and the AGPL-3.0 helps us dual-license our software to sell enterprise licenses. However, this will be expensive for customers. Licenses on the order of $100,000/yr will give customers little more than the enterprise license. Actual paid support for enterprise licenses is likely to cost on the order of $500,000/yr.

Because both of us have worked on a similar product (XTDB), we are acutely aware that consulting revenue is not product revenue. Enterprise licenses tend to blur this line and make it easy to bog down the company in support costs, while providing little to no advantage as a revenue model.

Growth Loop

We are also aware that, after 2024, we live or die by our early SaaS MRR. For Endb, the product *is* the growth loop. In 2024, we need to hook early customers with the easy start and SQL-native MongoDB replacement. Throughout 2025, we want to repeatedly deliver on our promise of "One Database" for existing customers, so they spend more on Endb.


In-place Analytics
Become the default home for data
Cheap read replicas
Customer wants to replace existing tools
Customer is happy with us as replacement
Infinite storage, temporal query
Easy start, dynamic SQL, exploratory queries

Example: Insurance Customer Adoption Journey

Susan is a semi-technical team lead at an insurance company. Legacy schemas evolve slowly, so she starts experimenting with Endb locally on her laptop. It's flexible, so some of her team follow suit. They decide to integrate multiple legacy schemas with Endb on-prem for a low-risk, high-value OpEx reporting project. Integration works well, so they start ingesting CDC from Accounting and Kafka events from the Quant/Risk team. Now Endb has a timeline of mutations. The team wants to build apps on Endb, so they switch to SaaS for ops and support. As more data moves into their fully-managed SaaS Endb cluster, management starts running analytics on this increasingly-useful dataset.

Investment

Our goal for this seed round is $3 million USD. We bootstrapped Endatabas throughout 2023. We are also the only contributors so far, though we do have a CLA in place.

With investment, we plan to hire 1 or 2 engineers in Europe (Håkan's timezone) and pay for cloud infrastructure costs to acquire initial customers. Our high-level operational goals are to remain as lean as possible; building a database is a huge undertaking and will take time.

Exit

Endatabas is a high-risk, high-reward exit: it targets the heart of the database market. Exit strategies include large vendor M&A (Oracle, Microsoft, IBM), service vendor M&A (Amazon, Google), or an IPO as early third-wave commercial open source.

Team

We have 20+ years of experience and have invested 6 years of R&D into Endatabas.

Prior to Endatabas, we both worked on XTDB, which was born from a bitemporal data lake Håkan built for a large bank's risk system in 2017. XTDB 1.x was developed from 2018 to 2020, with a focus on a Datalog query engine. XTDB 2.x was developed from 2021 to 2022, with a new focus on a SQL engine. Endb began in January of 2023. Although conceptually similar, Endb and XTDB 2.x have very different architectures; there is no shared IP and zero shared source code.

Prior to XTDB, Håkan ran a product team at ThoughtWorks Studios and Steven founded Nilenso, India's first worker-owned technology cooperative. Collectively, we've worked in nearly every major vertical. We met in 2007.

Appendix

Vision: Boot-to-Query in Under One Minute

We repeat this vision everywhere. JSON directly into the database in under one minute. Exploratory queries in under one minute. Combine heterogeneous schemas in under one minute. Beautiful error messages. Crystal clear documentation. All of this exists today.

Soon, it will also be possible for users to run arbitrary analytical queries in under one minute on their daily OLTP database.

Technology

Endb is written in Rust and Common Lisp. Steel Bank Common Lisp provides a battle-hardened dynamic runtime upon which to build Endb's dynamic SQL engine. Data is stored in Apache Arrow columnar format. Endb provides both HTTP and WebSockets APIs for stateless and stateful connections.

Regarding Endb's broader architectural choices, we have an extensive bibliography: https://www.endatabas.com/bibliography

Future Possibilities

The Endatabas architecture is amenable to all of the following enhancements. None of these features is a deviation from the core Endatabas vision:

- Vector Indexes
- Full-Text Search
- SQL:2023 (Property Graph Query)
- Bitemporal Indexes

References

i *It's the golden age of databases. It can't last.*, 2021, Joe Williams, Protocol, https://www.protocol.com/manuals/the-new-database/golden-age-databases-last

ii *What Goes Around Comes Around*, 2004, Stonebraker

iii *The Return of the General-Purpose Database*, 2021, Redmonk, https://redmonk.com/sogrady/2021/10/26/general-purpose-database

iv *Designing A Data-Intensive Future*, 2023, GOTO Unscripted, https://www.youtube.com/watch?v=P-9FwZxOIzE

v *Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases*, 2017, SIGMOD '17, page 1041–1052, Association for Computing Machinery.

vi *DBMS Market Transformation 2021: The Big Picture*, 2022, Merv Adrian, Gartner, https://web.archive.org/web/20221024090746/https://blogs.gartner.com/merv-adrian/2022/04/16/dbms-market-transformation-2021-the-big-picture/

vii *New Gartner report shows massive growth in database market, fueled by cloud*, 2022, Matt Asay, TechRepublic, https://www.techrepublic.com/article/new-gartner-report-shows-massive-growth-database-market-fueled-cloud/

viii *Cloud Storage Market Size, Share & Industry Analysis ... Forecast, 2024-2032*, 2023, Fortune Business Insights, https://www.fortunebusinessinsights.com/cloud-storage-market-102773

ix *Open Source Licenses for Developers*, 2023, Martin Clausen, https://www.youtube.com/watch?v=m478BHGR3XU