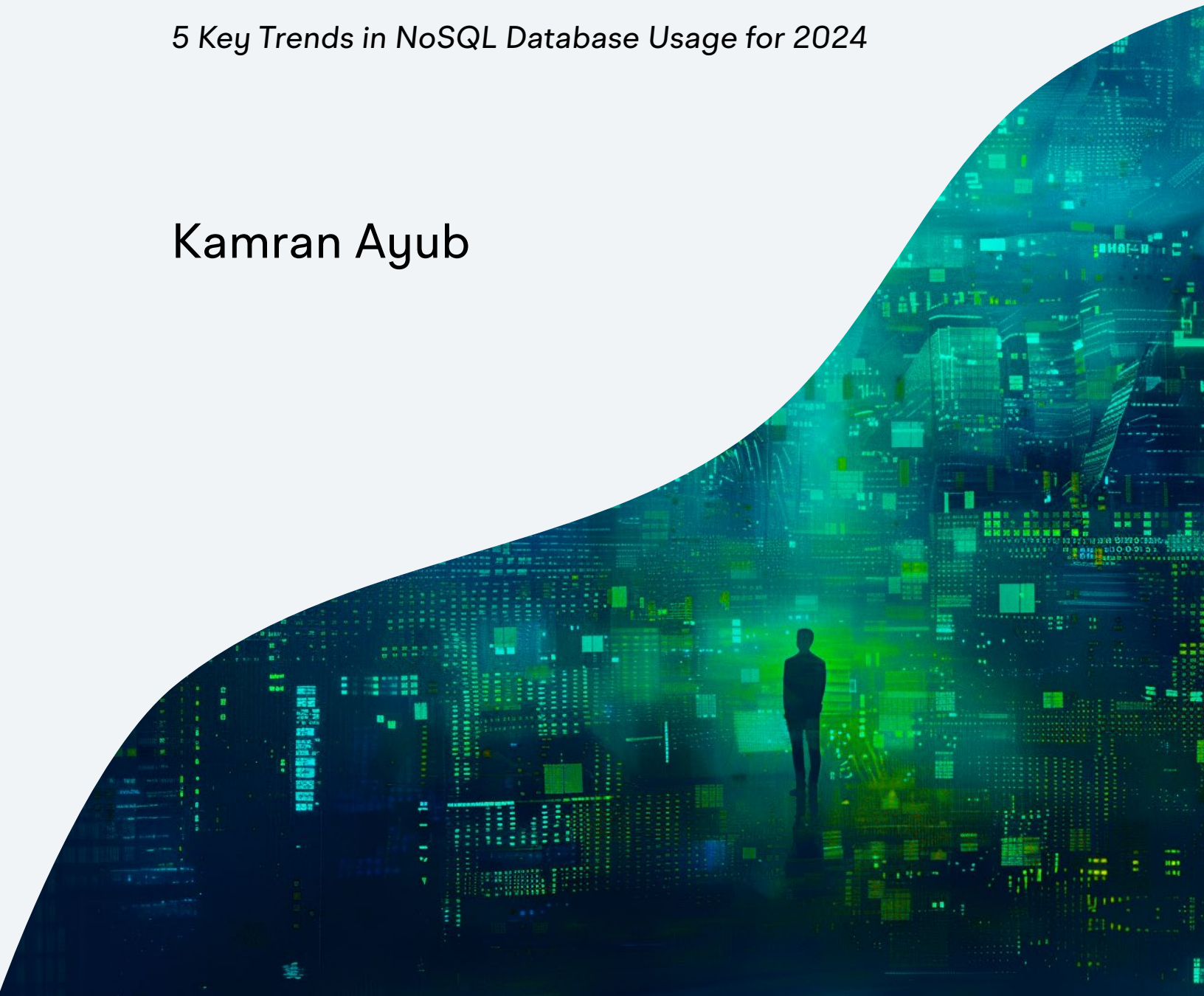
RavenDB

InfoQ Trend Report

5 Key Trends in NoSQL Database Usage for 2024

Kamran Ayub



Contents

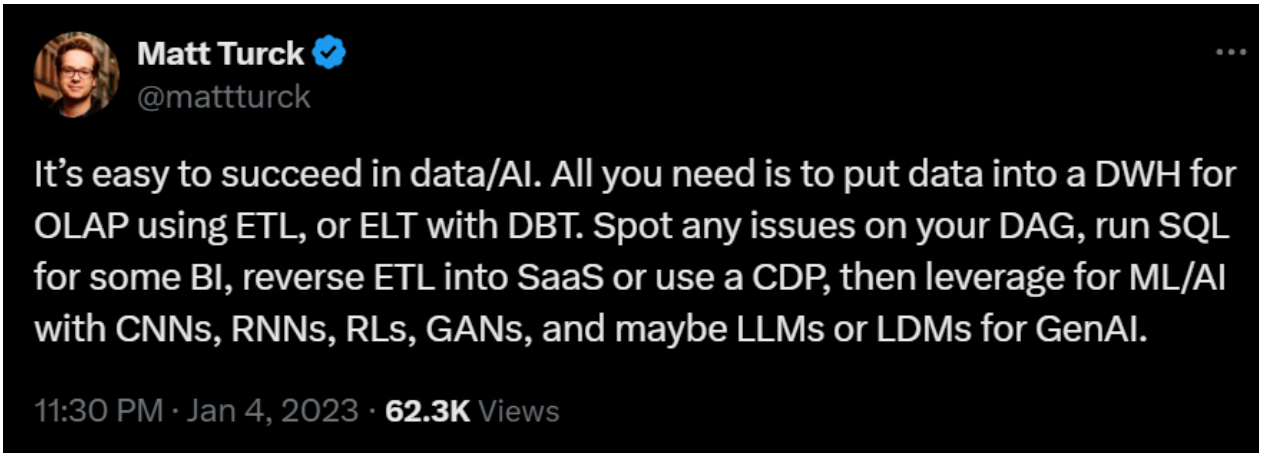
5 Key Trends in NoSQL Database Usage for 2024

In this report, we'll discuss 5 key trends we are seeing in the NoSQL space, drawing on data from the StackOverflow 2023 Developer Survey and JetBrains 2023 Developer Ecosystem. These developer surveys offer insights into how developers *use* databases with their technology stack, and that's what we'll focus on. We'll examine the current usage breakdown between relational and non-relational databases, highlighting where NoSQL solutions tend to fit into the picture.

With over 600 databases on the market, data and analytics leaders have their work cut out for them. Databases are core infrastructure, and organizations want to make sure they pick technologies that will be future-forward and, at the same time, compatible with their legacy systems. It's critical to understand how databases are used and with what technologies to enable better decision-making.

Defining NoSQL in the Modern Data Landscape

The complexity of the modern data landscape is, in a word, *overwhelming*, as summed up in this tweet by *Matt Turck*, VC at FirstMark:



Source: *@mattturck / X*

Matt and his team have been publishing the ML/AI/Data (or "MAD") Landscape for several years now to help make sense of it all:

Source: *Selection of databases from the MAD Landscape – a PDF also available*

Using this landscape, there are a few high-level categories of databases: relational (RDBMS), NoSQL, NewSQL, real-time, and graph. Within those categories, there's also a split by open source (OSS) vs. non-OSS. It's important to note that many databases under a single category can be "multi-model", that is, supporting multiple data modeling approaches like documents (RavenDB, MongoDB), relational (PostgreSQL), key-value (Redis), time series (RavenDB, InfluxDB), and graph (neo4j).

That is why the definition of "NoSQL" is fluid because more and more databases support hybrid features. Over the years, we've seen relational databases like PostgreSQL add NoSQL-like features, such as JSON data type, and NoSQL solutions like RavenDB adding time series support. Even more emergent databases like AWS VoltDB are categorized as "NewSQL" which offer flexible data modeling of NoSQL but with SQL querying support.

In this report, we've grouped survey responses for databases into two categories to try and keep things a bit simpler: *relational* and *non-relational*.

- **Relational:** Includes BigQuery, Clickhouse, Cockroachdb, DuckDB, Firebird, H2, IBM DB2, MariaDB, Microsoft Access, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, SQLite, Snowflake, Supabase

- **Non-Relational:** Includes Cassandra, Cloud Firestore, Cosmos DB, Couch DB, Couchbase, Datomic, Dynamodb, Elasticsearch, Firebase Realtime Database, InfluxDB, MongoDB, Neo4J, RavenDB, Redis, Solr, TiDB

2024 Key Insights

Before diving into the full report, here are the top-level insights we'll discuss:

1. Relational Databases Aren't Going Away (But Developers Don't Like Them): Nearly half (49%) of developers use a combination of RDMS and NoSQL together. Despite the rise of NoSQL and NewSQL databases, there is a consistent and strong reliance on traditional RDBMS systems like MySQL, PostgreSQL, and Microsoft SQL Server. That doesn't mean developers *like* it, though, with 27% of respondents indicating they want to stop using RDMS. Adoption across organization sizes is *eerily* consistent – from freelancers to the Fortune 500, everyone uses a consistent mix of RDMS and NoSQL solutions.

2. NoSQL Will Be Adopted for Specialist Roles: NoSQL databases like Redis and MongoDB have a high reciprocity with relational databases (between 50-69%, or well over half the time) indicating they are complementing rather than replacing relational databases. Even though some NoSQL databases like RavenDB *could* replace an RDMS application database (which it excels at), survey data makes it clear that most NoSQL solutions aren't used in a vacuum – they play a specialist role.

3. AI/ML Will Be a Major Driver of Data Demand: Developers are enamored with AI/ML but it isn't just excitement – they're already using it, with over 44% of professional developers indicating they've used AI/ML tools. Plus, 52% of data professionals increasingly use SQL databases as their data source when doing analysis.

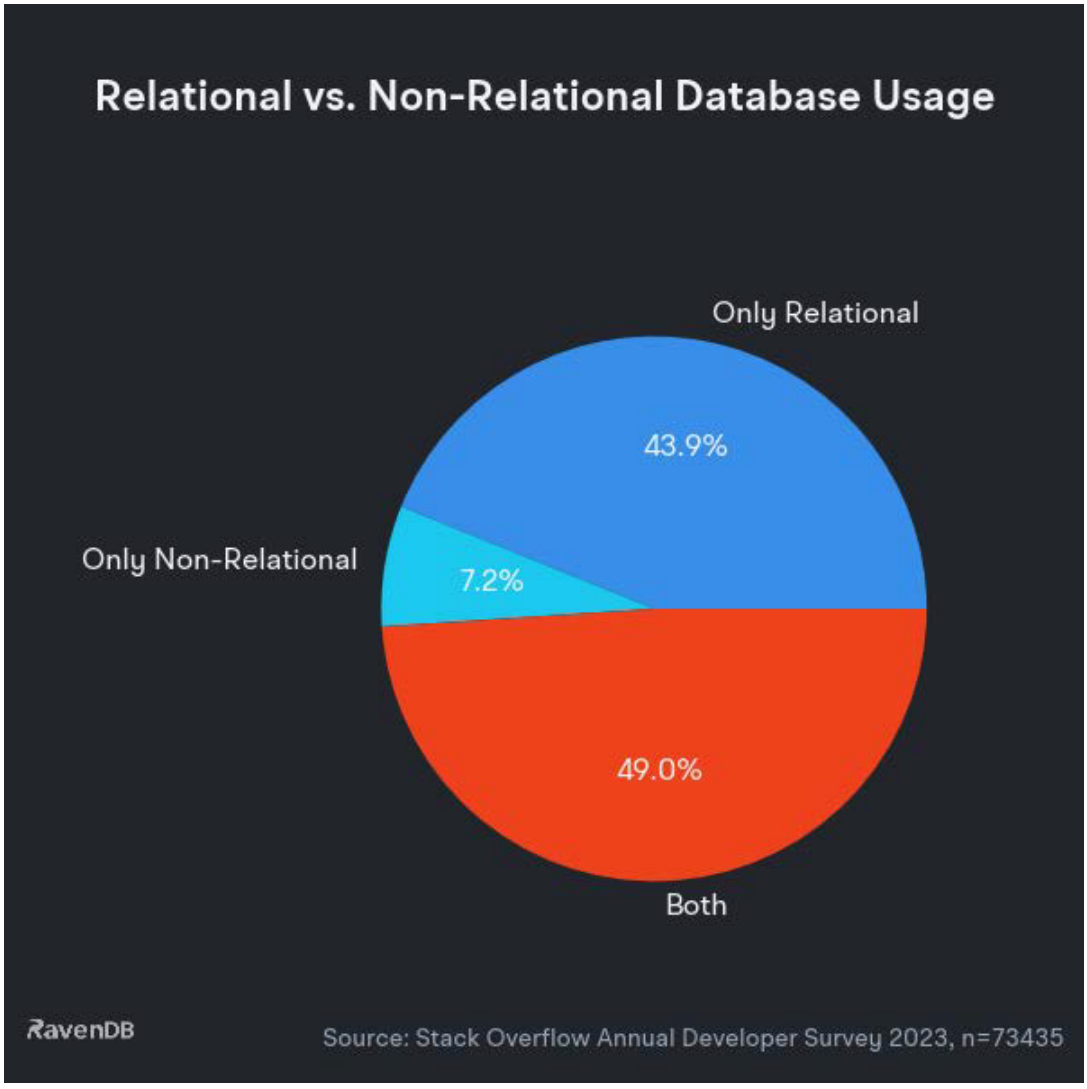
4. Rise of Niche Clouds and Managed Data: Overall, survey respondents showed a *marked* decrease in wanting to continue using the big three cloud providers (AWS, Azure, and Google Cloud). This could be due to the overall market climate, which is seeing tech companies scale back and cut costs. While cloud usage is growing overall, 45% of organizations still choose to self-host or use private cloud. However, the popularity of databases like BigQuery, Snowflake, and Cosmos DB, as well as "database abstractions" like Supabase (which is essentially PostgreSQL-as-a-service), points to a growing preference for cloud-native and managed database services. This shift may be driven by the desire for reduced overhead in database management and the benefits of cloud scalability and reliability.

5. NoSQL Will Increasingly Be Used With Distributed Systems. Infrastructure tooling is generally disliked by developers with most tools showing less desire to work with over the next year, but a few DevOps tools rise above the crowd: Kubernetes and Terraform. The desire to use these tools speaks to an increased prevalence of distributed system architecture and microservices. This is also reflected directly in past surveys, as organizations continue to adopt microservices and DevOps: 49% of respondents say they use microservice architecture, and 60% have a DevOps function in their organizations (both up 1% from 2022). We can also see that DevOps and build tools that remain popular with developers are more modern and offer better developer experiences.

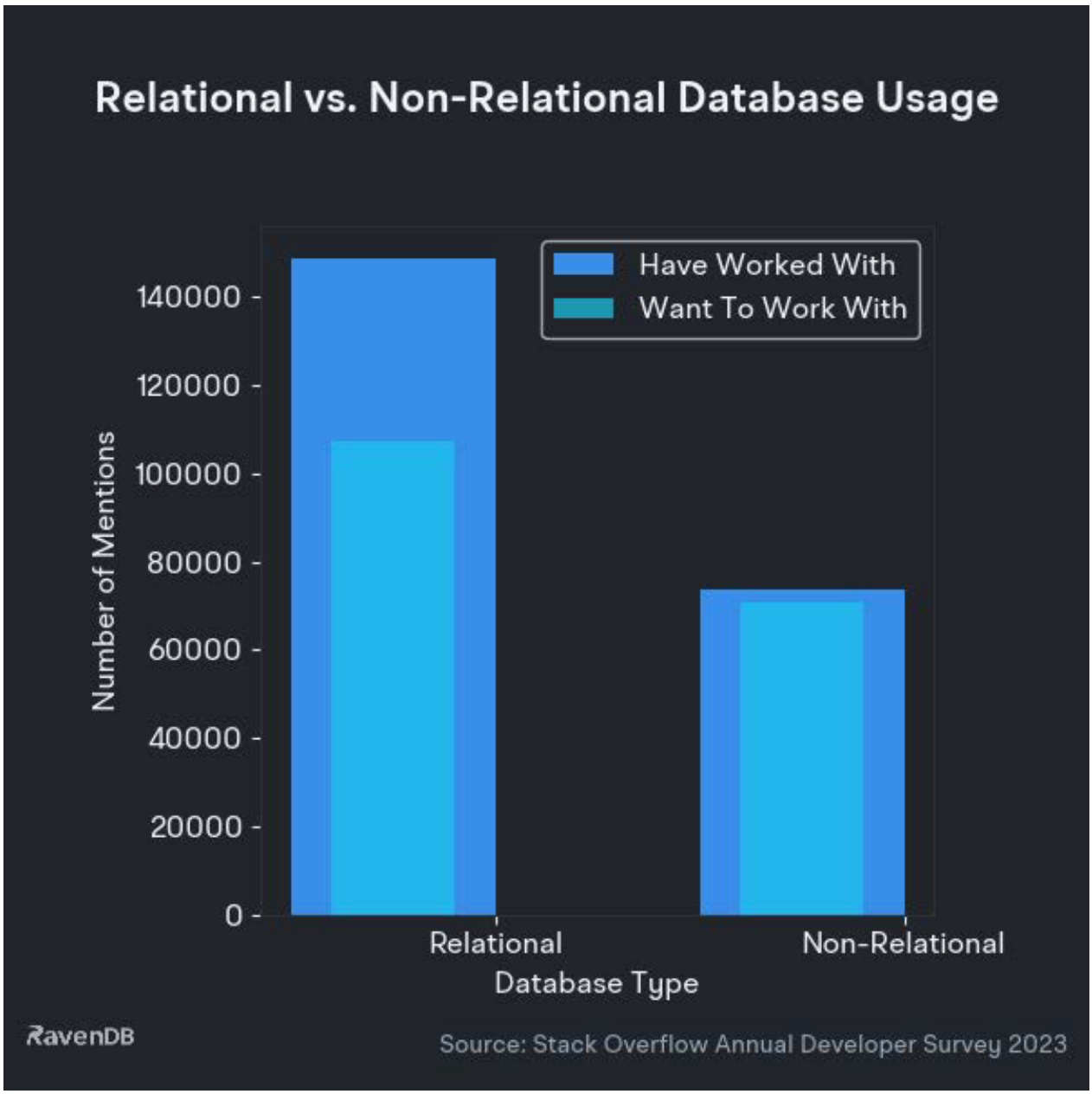
Relational Databases Aren't Going Away

(But Developers Don't Like Them)

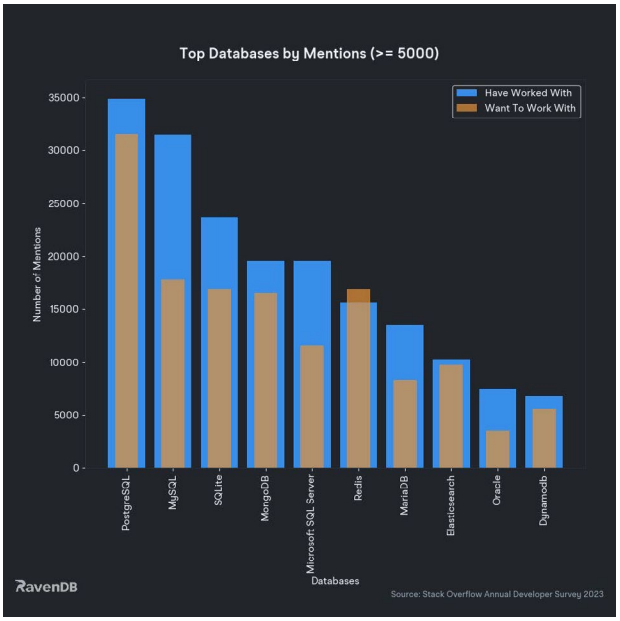
Since developers could choose *multiple* options for what databases they've worked with (or want to work with) in the past year, we can better understand NoSQL usage by grouping respondents to those who *only* use relational databases, *only* use non-relational databases, or use *both*.

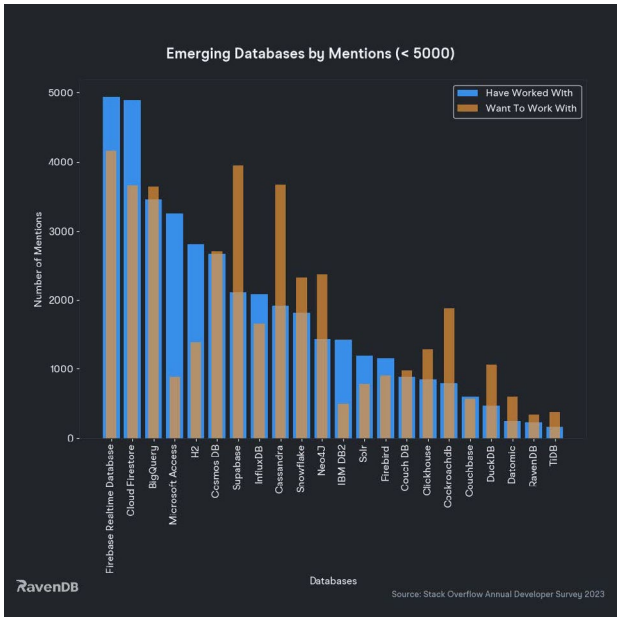


Relational Databases Remain a Constant. While 43% of respondents say they exclusively use RDMS, 49% of them also use NoSQL. It shouldn't be too surprising to learn that less than 8% only use NoSQL – with their entrenched usage and such an emphasis on data analysis and reporting, it's hard to escape the relational database orbit. Developers don't necessarily like it though – when we analyze what database type developers want to work with, there's a significant amount of survey respondents (27%, or almost a third) who indicate they don't want to use RDMS in the future which speaks to an increasing dissatisfaction with RDMS solutions overall. On the flip side, developers who use NoSQL seem to be much happier and prefer to keep using it.



This isn't the whole story though – not all relational databases are hated. If we dive deeper into the responses for database usage, we can see a few interesting trends that explain the overall sentiment:





Everyone Loves PostgreSQL. PostgreSQL remains the most used RDMS and also the most loved – with relatively few respondents saying they want to stop using it. This could explain why there's major interest in Supabase (an open source Firebase alternative offering Postgres-as-a-Service). Instead, where we see the biggest dissatisfaction is in older RDBMS like MySQL, SQL Server, or Oracle.

NoSQL is Capturing Interest. On the other hand, the survey paints a bright picture for new or alternative entrants to the database space with surging interest in over 14 products. It's clear that developers are tired of the "old guard" and looking for new solutions. Some key NoSQL databases developers will be looking at using more in 2024 include: RavenDB, Redis, BigQuery, Supabase, Cassandra, Snowflake, Neo4j, Clickhouse, Cockroachdb, DuckDB, Datatomic, and TiDB.

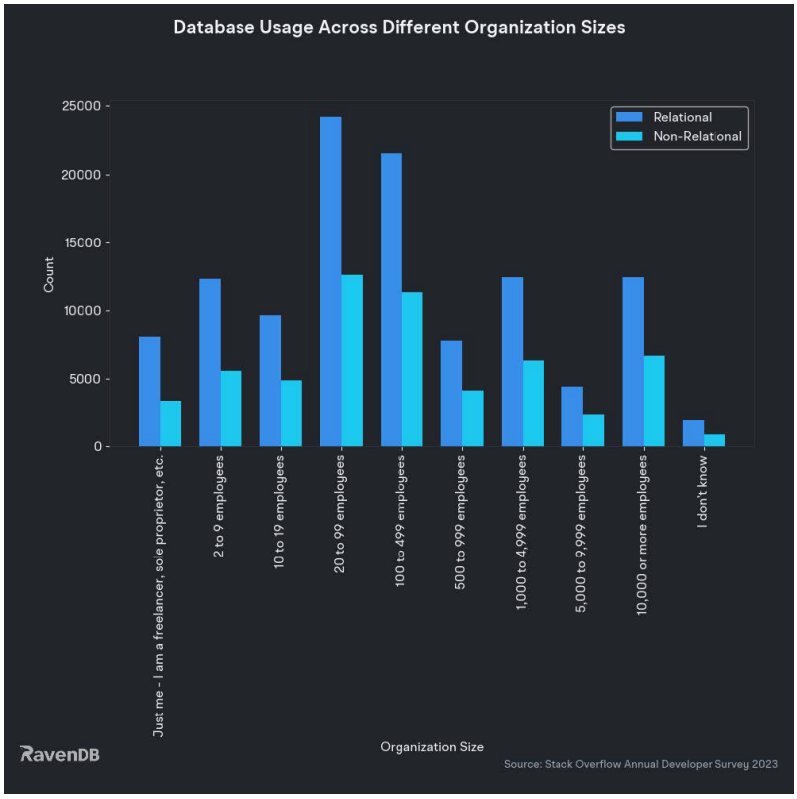
RavenDB Is Like SQL's Cooler, Younger Cousin

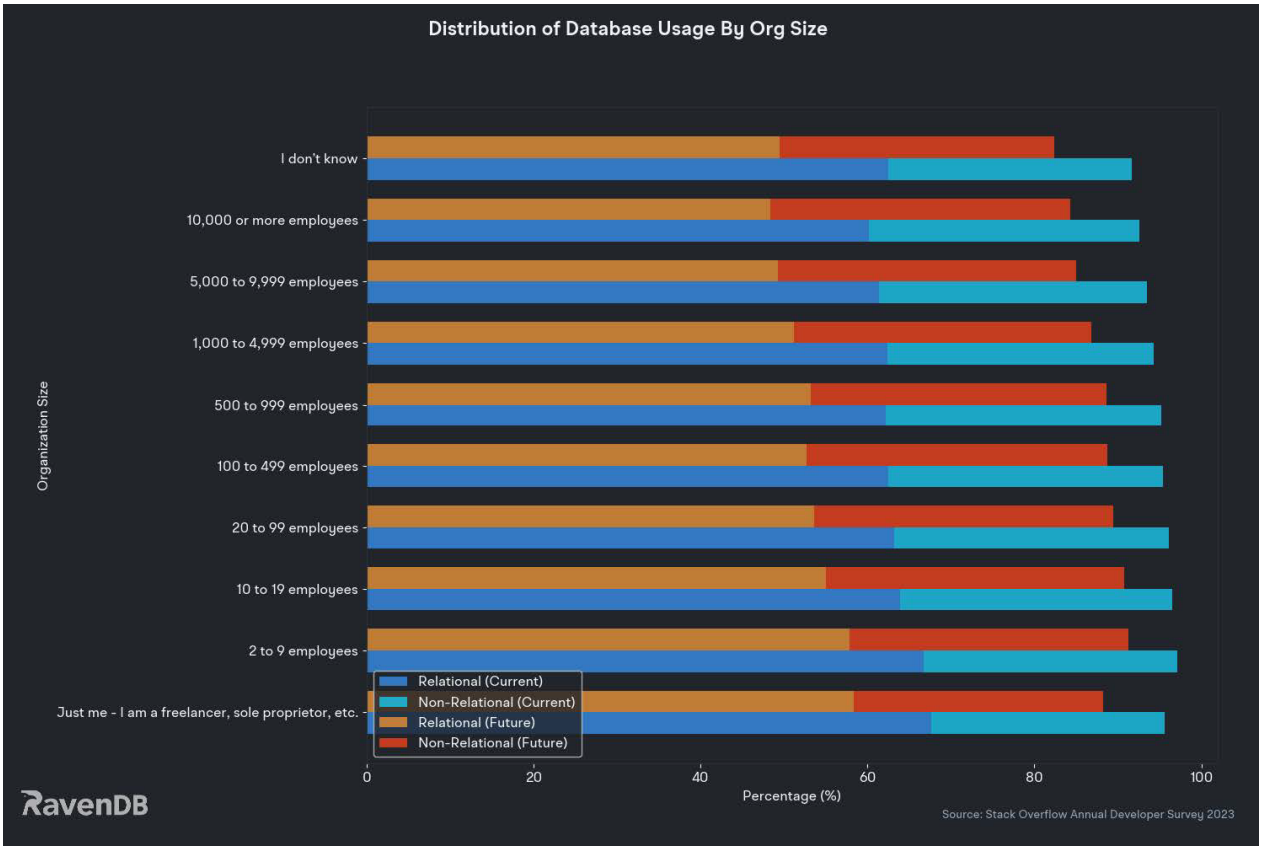
RavenDB is a multi-model document database that is designed to be used alongside RDMS solutions. It features SQL ETL and SQL Replication as well as some, gasp, relational capability in queries. It's not SQL but it moves the most expensive part of relational databases, JOINs, from runtime to its ahead-of-time compiled indexes. You can use RavenDB as a primary application database and automatically replicate data to an RDMS for reporting & analytics with support for PostgreSQL and SQL Server.

NoSQL Will Be Adopted for Specialist Roles

Orgs Use a Steady Mix of RDMS and NoSQL. This may be surprising, but database preferences and usage don't appear to differ much across org sizes. Everyone across every org size generally wants to move away from RDMS to NoSQL. In other words, smaller companies are not predominantly leaning towards one type of database over another in comparison to larger companies, and vice versa. This could imply that the choice of database technology is influenced more by the technology's features and suitability for specific tasks rather than by the size of the organization.

The uniformity might also indicate a technology-agnostic approach in the industry, where decisions are based more on the specific requirements of a project rather than any prevailing industry-wide trend towards a certain type of database.





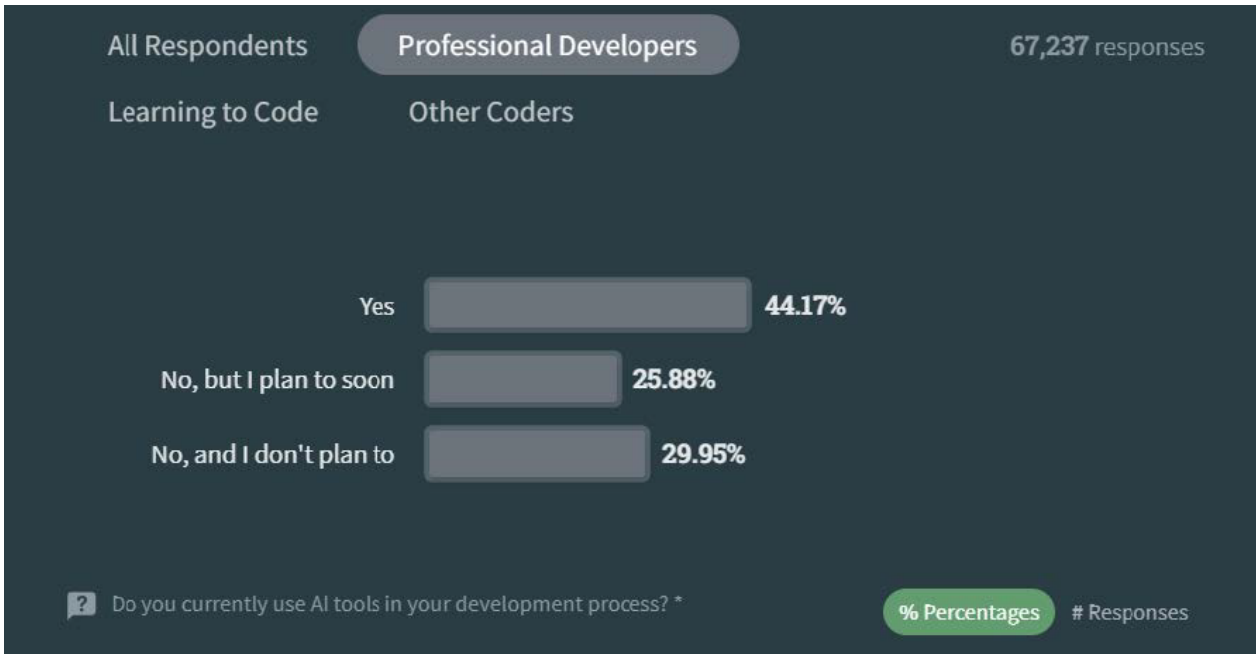
Emergence of Specialized NoSQL: Less than 8% of developers *exclusively* use NoSQL which speaks to an overall need to be interoperable with RDMS. The presence of specialized databases like InfluxDB (time-series) and Neo4J (graph database) in the Non-RDBMS category highlights the growing niche areas in database technology. These specialized databases are becoming more important as applications become more complex and require specific types of data processing.

NoSQL Has High Reciprocity with RDMS: Another reason to believe NoSQL will continue to be used alongside RDMS is when analyzing "reciprocity" (that is, developers who use A *with* B). We can see that NoSQL solutions aren't being used to necessarily *replace* traditional RDMS but to *augment* solutions. The two most popular NoSQL databases, Redis and MongoDB, show between 50-69% reciprocity with popular RDMS databases like MySQL and PostgreSQL. This implies that organizations tend not to use NoSQL in a vacuum but as a way to serve specialized use cases and then integrate them with their RDMS, possibly for analytics/reporting.

MySQL	PostgreSQL	MongoDB	Redis	SQLite	MS SQL Server	MariaDB	Oracle Database	
–	51%	64%	69%	59%	53%	75%	58%	MySQL
38%	–	50%	55%	51%	39%	50%	40%	PostgreSQL
33%	35%	–	40%	32%	31%	37%	29%	MongoDB
35%	37%	39%	–	34%	22%	38%	29%	Redis
29%	33%	30%	32%	–	28%	38%	21%	SQLite
18%	18%	21%	15%	20%	–	20%	29%	MS SQL Server
23%	20%	22%	23%	24%	18%	–	16%	MariaDB
14%	13%	14%	14%	11%	21%	13%	–	Oracle Database
3%	3%	3%	3%	2%	4%	3%	6%	DB2
3%	4%	5%	5%	2%	3%	3%	4%	Cassandra
1%	1%	1%	1%	1%	1%	1%	2%	Couchbase
2%	2%	2%	4%	2%	2%	2%	3%	HBase
3%	3%	4%	4%	3%	2%	3%	3%	Neo4j
3%	5%	5%	4%	3%	5%	4%	6%	Amazon Redshift
7%	9%	8%	9%	6%	6%	10%	12%	H2
3%	4%	3%	6%	2%	2%	3%	3%	ClickHouse
5%	8%	6%	7%	6%	7%	6%	6%	Other

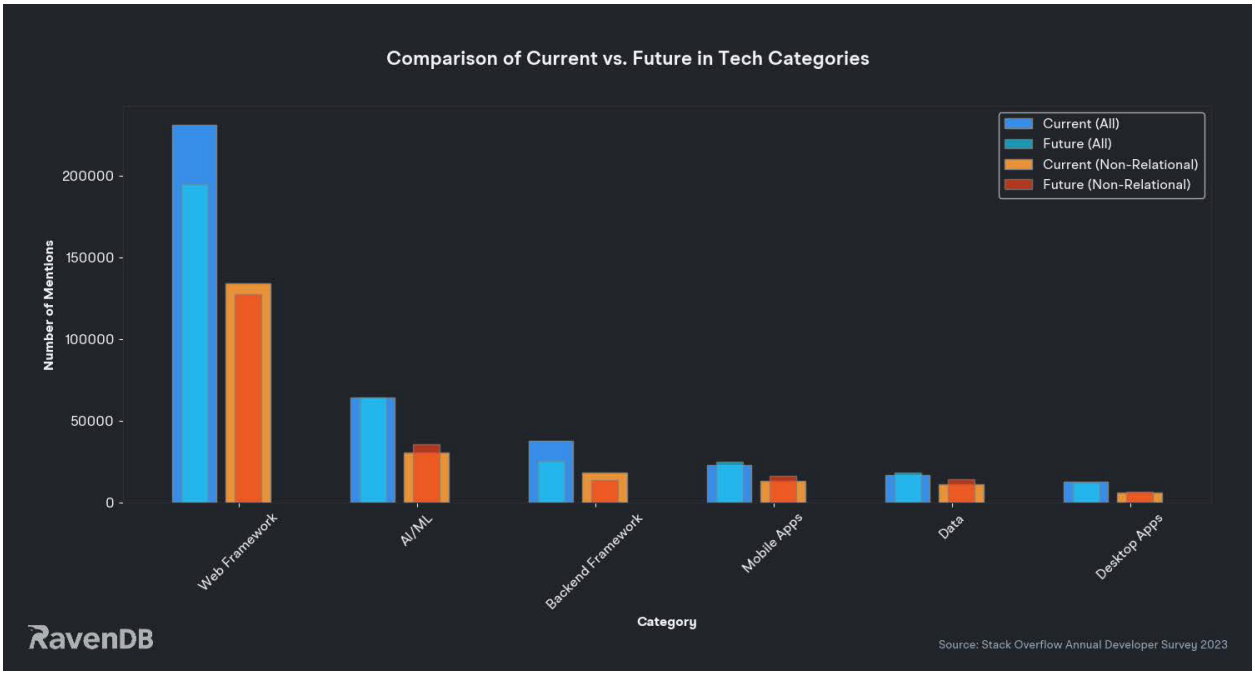
Source: *Databases*, ***JetBrains Developer Ecosystem 2023***

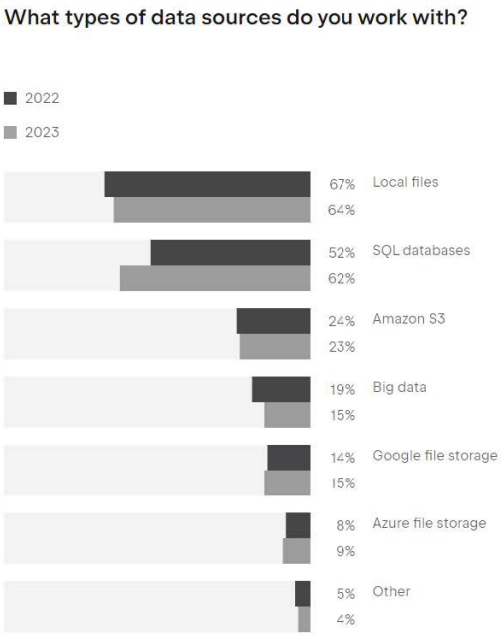
AI/ML Will Be a Major Driver of Data Demand



With over 44% of professional developers reportedly using or planning to use AI/ML tools in the next year, this use case will be a primary driver of data needs – for NoSQL and RDMS alike.

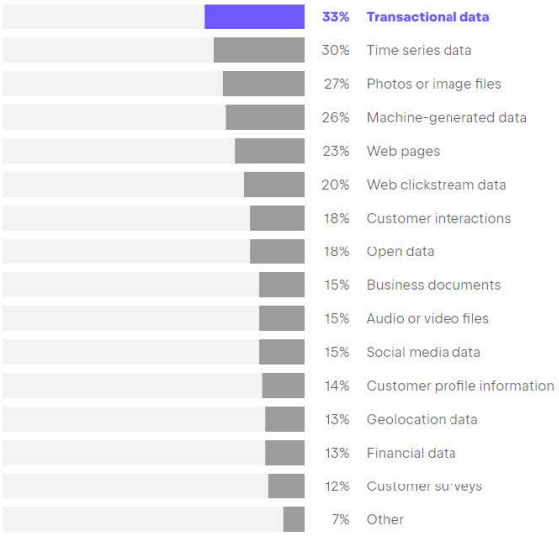
AI Use Cases Will Continue to Grow. When we group the unique responses to "Miscellaneous Tool" usage in the StackOverflow developer survey, we find the AI/ML category outstrips all other miscellaneous categories by more than 2X (Web Frameworks is its own separate category). Developers who want to use more NoSQL technologies have slightly more interest in adopting AI/ML tech than other developers. Developers who are already using AI/ML tools want to continue using them. 47% of respondents to the JetBrains survey indicated they are building AI-based features into products and services, with an increasing amount (52%) relying on SQL databases for their data sources.



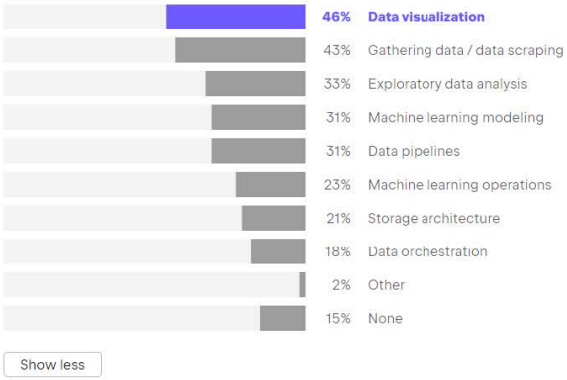


Although the majority uses local files, the share of those using SQL databases grew by 10 percentage points over the past year, highlighting the importance of SQL for data science.

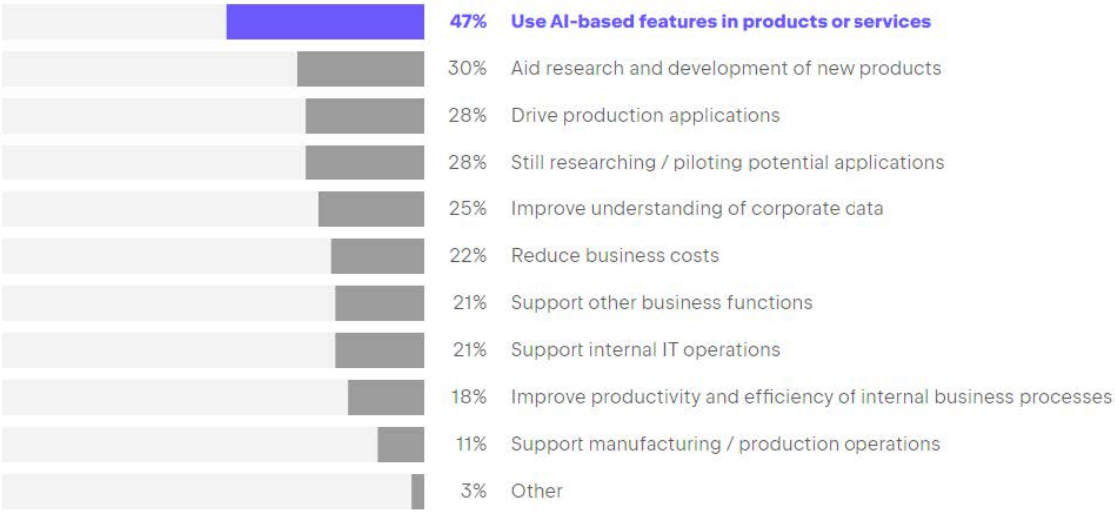
What types of data do you analyze?



In which of the following activities are you involved?



Which of the following best describes the use of machine learning in your organization?



Source: *Data Science* ***– JetBrains Developer Ecosystem 2023***

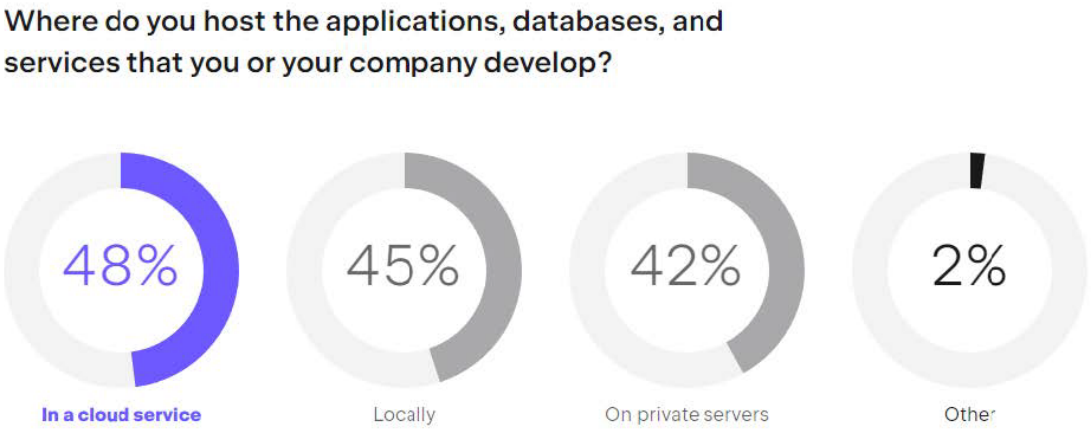
Growing Mobile and Data Integration Needs. There is growing interest across the board in native mobile application development (e.g. React Native, Flutter, MAUI, Ionic) and data tools (e.g. Kafka, RabbitMQ, Spark). After "transactional data," respondents to the JetBrains survey listed "time series data" and "clickstream data" in their top 10 types of data they analyze. Specialized NoSQL databases like Clickhouse (clickstream) and InfluxDB (time series) excel at collecting this data, but analysis tends to happen with SQL indicating a growing need for ETL solutions with streaming data tools like Kafka and RabbitMQ.

Use Cases Drive Database Adoption. Category trends are the same across all respondents, regardless of interest in database technology – this further adds to our understanding that developers expect NoSQL databases to work for the same use cases they use RDMS technologies.

Rise of Niche Clouds and Managed Data

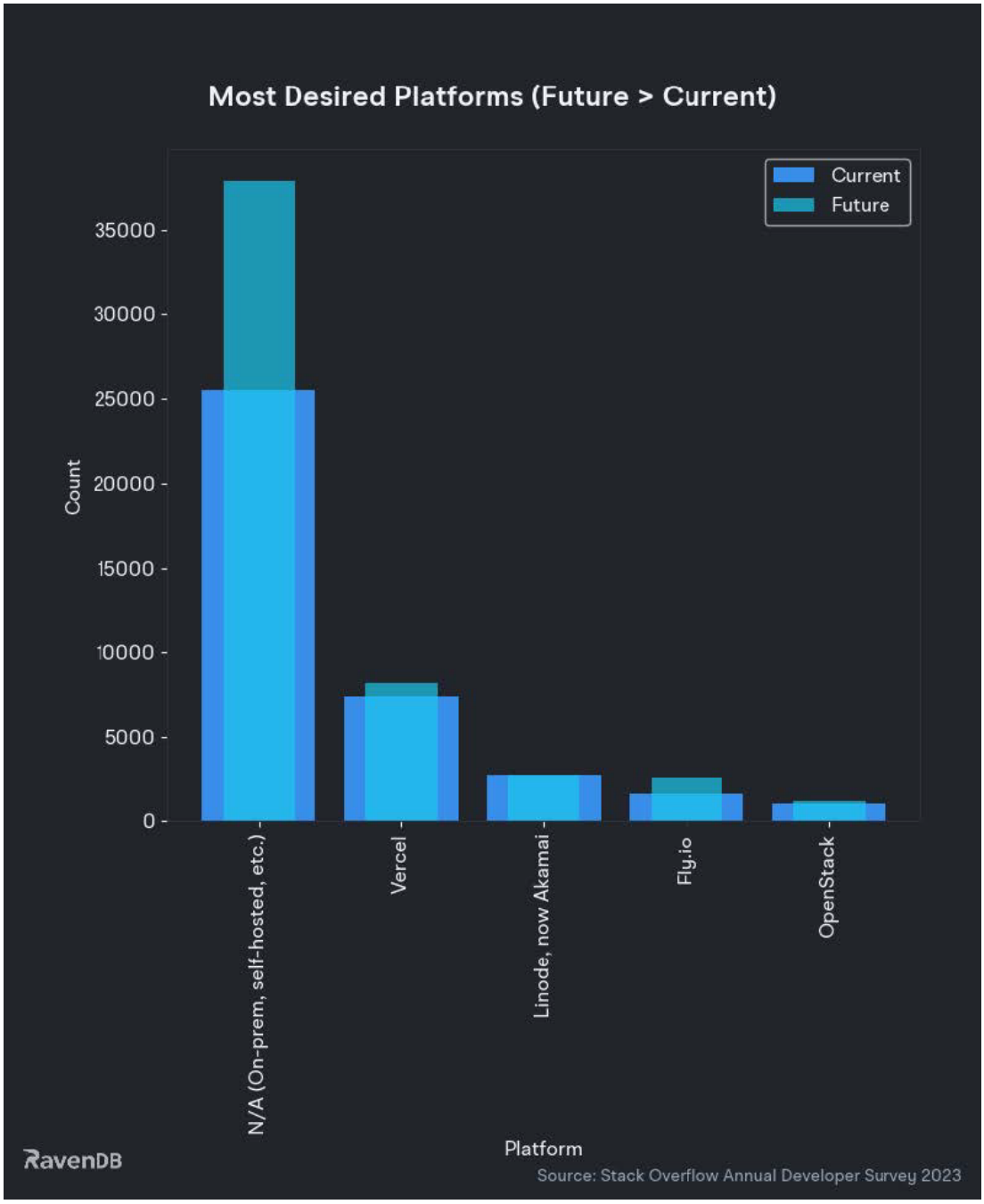
While organizations continue the march toward moving to the cloud, with over 48% of survey respondents saying they host their applications in a cloud service, from a developer perspective, there seems to be growing dissatisfaction with the big three (AWS, Microsoft Azure, and Google Cloud) and a big desire to self-host. There's also some growing interest in "niche" clouds like Cloudflare, Fly.io, and Vercel, which offer more tightly integrated services or edge hosting capabilities.

Data Solutions Will Need to Bridge the Hybrid Cloud. According to the JetBrains survey, 44% of services are hosted primarily in a hybrid fashion with a mix of cloud and private servers. This still leaves the majority of organizations choosing not to use the cloud. Organizations moving to the cloud will need data products that can still operate on-premise, spanning the mix of private vs. public cloud.



Source: DevOps – JetBrains Developer Ecosystem, 2023

Is Cost-cutting Putting the Brakes on Cloud Adoption? In the StackOverflow developer survey, the three major clouds (AWS, Azure, Google) are showing some churn when it comes to developer sentiment with a heavy "want" skew towards "N/A" responses – which we can take as wanting to self-host or use private servers. Interestingly, NoSQL developers use platforms more heavily than the general population and still would prefer to outsource hosting (likely due to the horizontal scaling nature of NoSQL workloads which is harder to manage on your own).



Rising Usage of Edge Computing (and Consequently, Data). With increasing interest in edge-friendly clouds like Cloudflare, Vercel and Fly.io, developers seem to be interested in taking advantage of closer end-user deployments. Many of these platforms offer their own hosted data solutions with a combination of RDMS (Vercel Postgres, Cloudflare D1) and non-RDMS (Cloudflare KV/R2, Vercel Object) storage options that feature edge-replication.

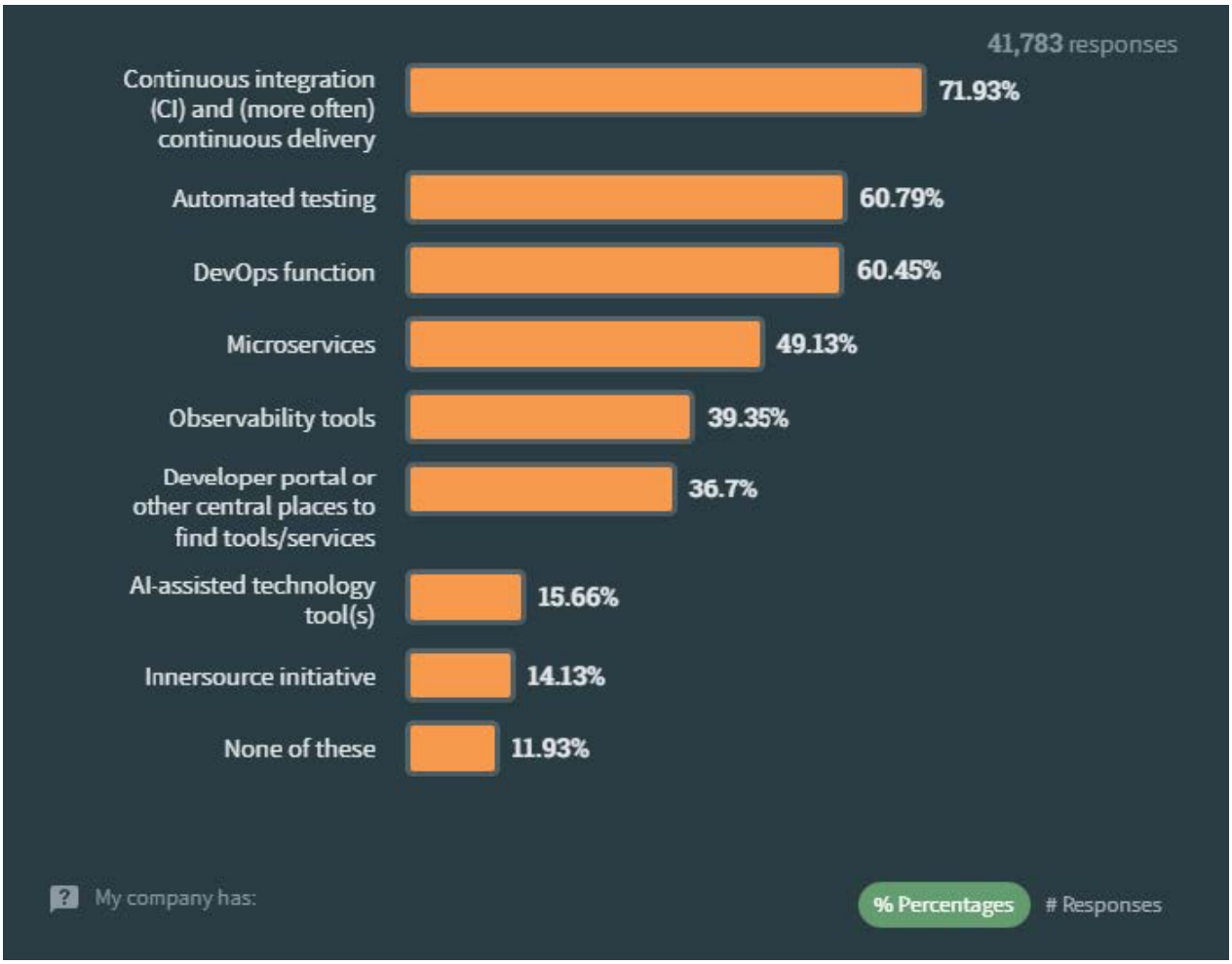
Better Open Source Tooling Makes Self-Hosting More Palatable. With developers happy with and wanting to adopt tools like Docker and Kubernetes, could this explain why so many want to move away from public cloud? Or is it just because the AWS Console is *notoriously terrible*? Perhaps developers believe it's worth the elbow grease to use Terraform to manage their own cluster and save on fully-managed service costs.

RavenDB Supports Self-Hosting, Cloud, or Both

RavenDB Cloud's database-as-a-service (DBaaS) offers multi-cloud managed hosting on AWS, Azure, and Google Cloud, which is cheaper than managing compute instances yourself. For on-premise or private servers, RavenDB lets you choose to self-host in a standalone fashion on Linux/Windows or with Docker. Clusters can span across regions, availability groups, or public and private clouds, making it the perfect option for hybrid cloud solutions. Also unlike AWS, RavenDB offers a modern Studio interface.

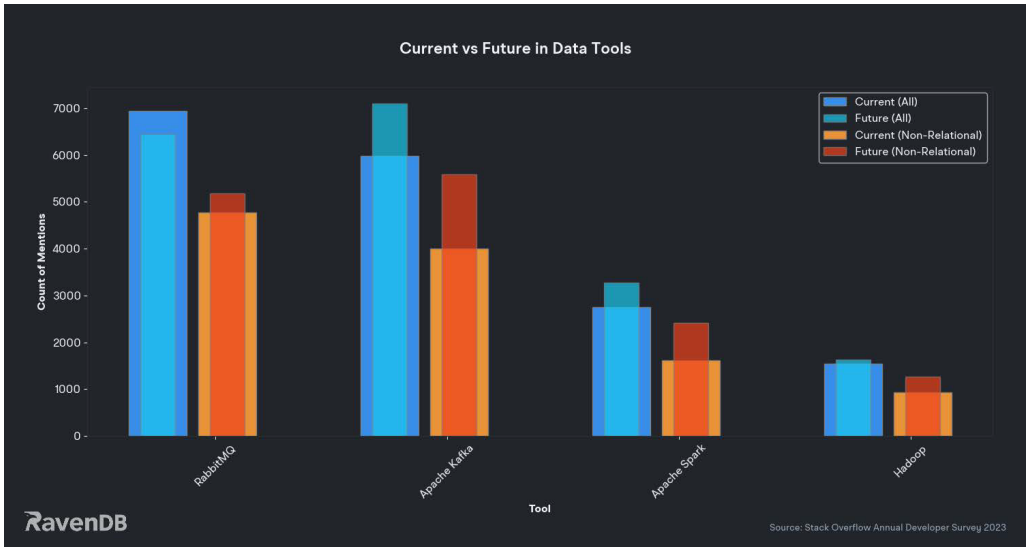
NoSQL Will Increasingly Be Used With Distributed Systems

In the StackOverflow survey, respondents were asked what kinds of processes, tools, and programs were in place at their organizations. We see an increasing trend in organizations adopting modern development practices like creating a DevOps function, using CI/CD pipelines, and implementing more distributed system architectures like microservices.

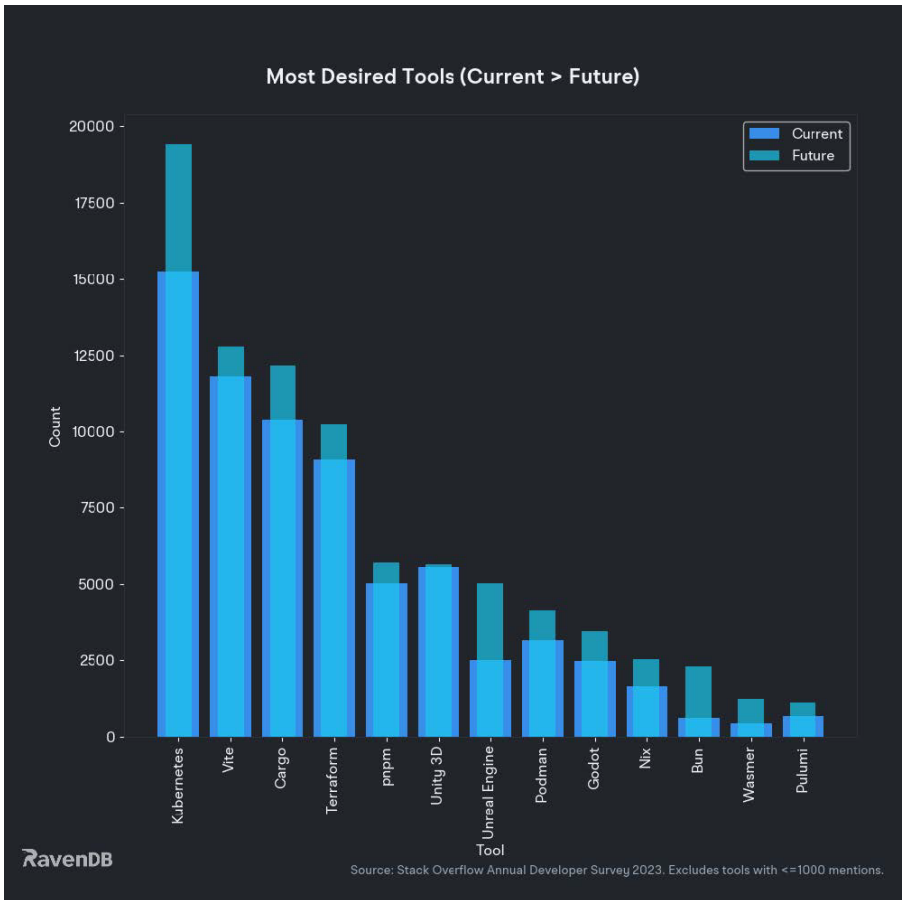


Source: ***Responses to "****Developer Experience: Processes, tools, and programs within an organization****"***
–StackOverflow Developer Survey 2023

We can analyze the responses to the "Tools and Tech" survey question further, separating NoSQL developers from the general population. This overarching trend is also reflected in individual tools, as we can also see increased interest in streaming data tools like Kafka and RabbitMQ, as well as "Big Data" tools like Apache Spark and Hadoop:



Software Developers Generally Dislike Infrastructure Tools. There's a general distaste for infrastructure tools across the board with the majority of tools showing decreased interest in "wanting to use" next year, whether it's frontend or backend infrastructure. This makes sense as wrestling with tooling takes away from working on features and software development activities.



More Kubernetes, Please. While developers are looking to use less Docker, they seem to want to use more Kubernetes (or "k8s" for short). This could be a symptom of Docker's recent *Docker Desktop licensing changes*, and a move to more open source friendly tooling. Since Kubernetes is designed for scaling and orchestrating container infrastructure, this could drive more demand for NoSQL solutions that are designed for better versioning for modern DevOps workflows since databases generally are *difficult to version*.

Tooling Will Become Easier and More Streamlined. Modern tools like Vite, Terraform, Cargo, pnpm, and Podman all offer a better developer experience, performance, and less friction than the tools they are meant to replace. This indicates developers will want to see increasing focus on developer experience.

Looking Ahead to 2024

As we encapsulate the insights from our comprehensive analysis of the NoSQL landscape, it's clear that the ecosystem is poised for significant evolution. The trends towards cloud-native solutions, real-time data processing, and flexible data models underscore a paradigm shift in database technologies. NoSQL, with its inherent scalability and versatility, is at the forefront of this transformation.

- **NoSQL's Broader Implications:** The increasing adoption of NoSQL databases reflects a deeper change in how data is managed and utilized. Businesses are seeking solutions that offer not just storage, but also analytical capabilities, real-time processing, and seamless integration with other technologies. This shift is driving innovation across various sectors, influencing how data-driven decisions are made.

- **The Role of Cloud and Managed Services:** Cloud-native and managed database solutions have emerged as key players, offering scalability and efficiency. This trend highlights the industry's move towards more agile, service-oriented database management, reducing the complexity and overhead associated with traditional database systems.

- **Adapting to Market Demands:** As the gap between current and desired database knowledge widens, there's an opportunity for platforms like RavenDB to lead in education and skill development. By addressing this skills gap, RavenDB can not only enhance its user base but also contribute to shaping the future of database technology.

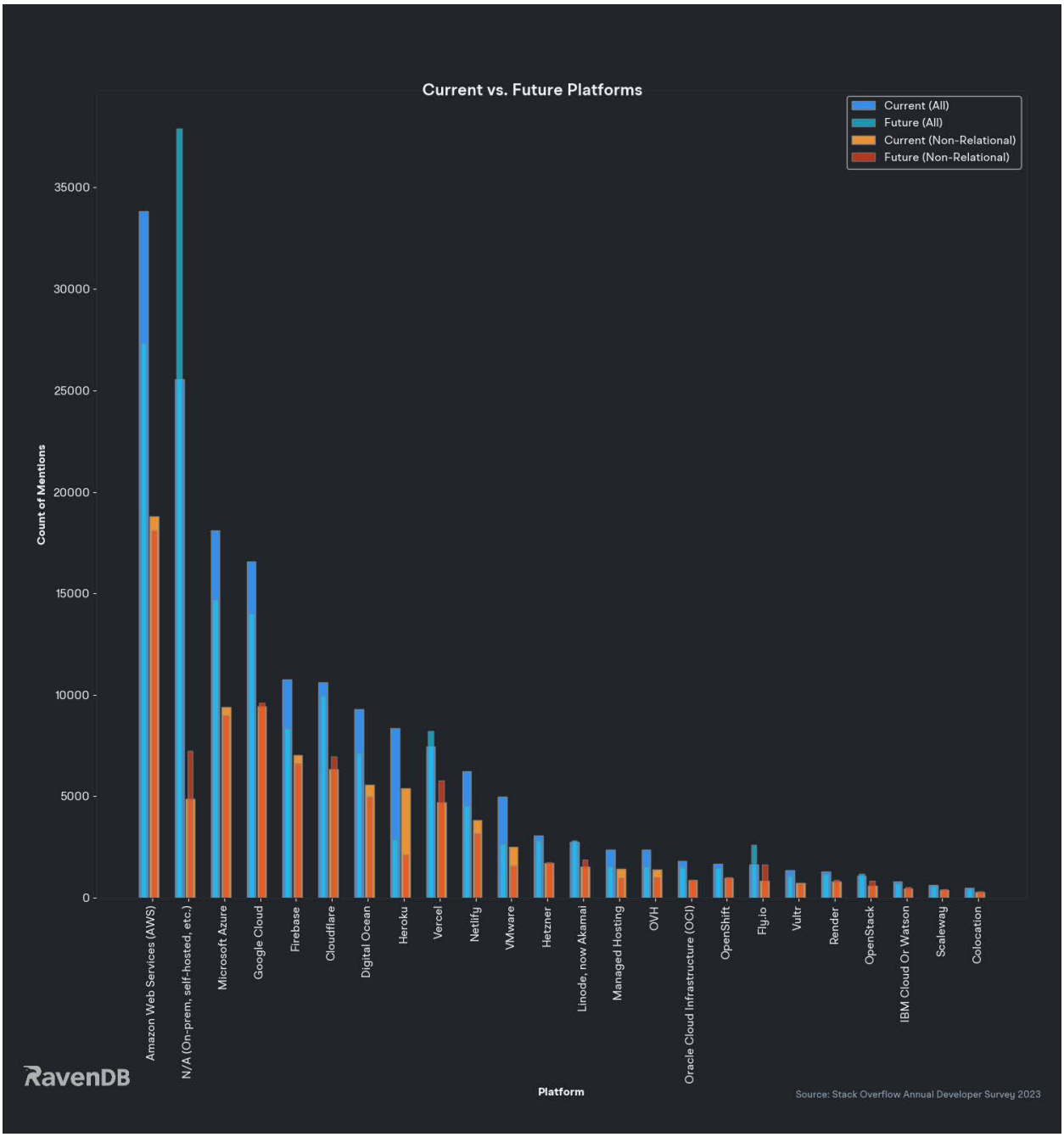
- **Strategic Recommendations for Industry Players:** For businesses and developers navigating this landscape, embracing NoSQL solutions like RavenDB is not just about adopting a new technology—it's about investing in a future-proof, adaptable data architecture. The key is to stay informed, adaptable, and open to the evolving possibilities that NoSQL databases offer.

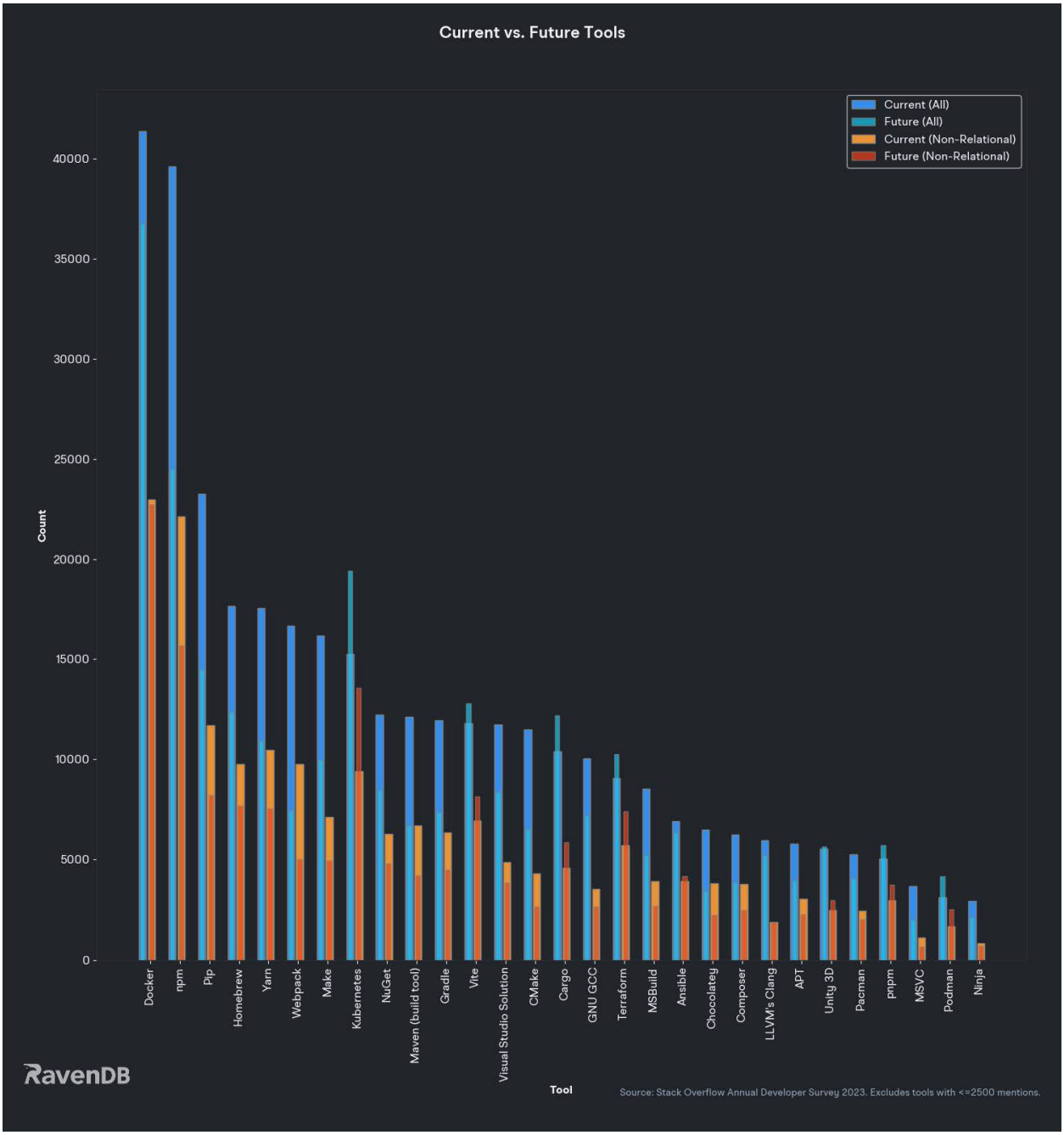
RavenDB: A Future-Forward Choice

In this evolving landscape, RavenDB stands out as a particularly future-forward option. Its multi-model approach, offering document, object, and time-series databases in one package, aligns perfectly with the diverse needs of modern applications. RavenDB's emphasis on high availability, robust performance, ETL interoperability, and developer experience makes it a compelling choice for businesses looking to leverage NoSQL technologies.

In summary, the NoSQL domain, led by innovative solutions like RavenDB, is transitioning into a more dynamic, efficient, and scalable era. This evolution presents exciting opportunities for businesses and developers alike to redefine their data strategies and stay ahead in a data-centric world.

Appendix: Supporting Materials




Current vs. Future Tools
Current (All)
Future (All)
Current (Non-Relational)
Future (Non-Relational)
Count
40000
35000
30000
25000
20000
15000
10000
5000
0
Docker
npm
Pip
Homebrew
Yarn
Webpack
Make
Kubernetes
NuGet
Maven (build tool)
Gradle
Vite
Visual Studio Solution
CMake
Cargo
GNU GCC
Terraform
MSBuild
Ansible
Chocolatey
Composer
LLVM's Clang
APT
Unity 3D
Pacman
pnpm
MSVC
Podman
Ninja
Tool
RavenDB
Source: Stack Overflow Annual Developer Survey 2023. Excludes tools with <=2500 mentions.

Kamran Ayub is a developer, educator, speaker, and the founder of Keep Track of My Games. He teaches developers modern web development, cloud, and NoSQL as a Pluralsight author. He also helps maintain the Excalibur.js game engine. Previously, he helped build and scale massive websites for Fortune 500 companies.

About RavenDB

RavenDB is a pioneer in NoSQL database technology with over 2 million downloads and thousands of customers from startups to Fortune 100 Large Enterprises.

Mentioned in both Gartner and Forrester research, over 1,000 businesses use RavenDB for IoT, Big Data, Microservices Architecture, fast performance, a distributed data network, and everything you need to support a modern application stack for today's user.

For more information please visit

ravendb.net

Contact us at

info@ravendb.net

Documentation
https://ravendb.net/learn/docs-guide

Use Cases
https://ravendb.net/news/use-cases

Free Online Training
https://ravendb.net/learn/bootcamp

Webinars
https://ravendb.net/learn/webinars

RavenDB Download
https://ravendb.net/download

RavenDB Cloud Database as a Service
https://cloud.ravendb.net/


RavenDB