



! 5 . / . οΩ° σ / ůΨμůσ†ŏΠΡούβι ůŮ
 ΛΩŚŸσúοβι [§†ŚΩ° 5†ú† £ŮŮΡσ
 {ú†ŌΞσ

Thank you for downloading this Voltron Data resource. Carahsoft is the distributor for Voltron Data's AI and ML solutions available via NASA SEWP V, ITES-SW2, NASPO ValuePoint, and many more contract vehicles.

To learn how to take the next step toward acquiring Voltron Data's solutions, please check out the following resources and information:



For Voltron Data overview:
carah.io/Voltron-Data



For upcoming events:
carah.io/Voltron-Events



For additional resources:
carah.io/Voltron-Resources



For additional Artificial Intelligence:
carah.io/ai-solutions



To set up a meeting:
VoltronData@carahsoft.com



To purchase, check out the contract vehicles available for procurement:
carah.io/procurement

ADBC Brings Composability to Industry Leading Data Tools, Stacks

Kae Suarez



Voltron Data helped introduce and actively contributes to the Arrow Database Connectivity (ADBC) standard to enable existing clients and servers to speak to each other using columnar data. Since the [first release in January](#), we've been working to ensure portability and functionality across any system that benefits from columnar data communication. This gives users massive advantages in terms of data transfer speed and data analytical processing throughput.

Recently, the wider data analytics community has taken note — three leading technology providers have adopted ADBC. We're excited by this progress and how it supports data analytics software stacks. In this post, we'll give you a primer on ADBC and cover how industry-leading organizations have integrated ADBC into their stack.

What is Arrow Database Connectivity (ADBC)?

ADBC, much like JDBC and ODBC, is an API standard that can be implemented via drivers allowing generic communication to different server backends. Where ADBC provides an advantage is in its emphasis on bulk columnar data retrieval and ingestion — where JDBC and ODBC are row-based. This is especially vital in the modern era of data analytics because many systems, like those we'll discuss today, are columnar behind-the-scenes to enable faster analytics. ADBC ensures that

there's no bottleneck from converting to and from row-based data for transmission, like with JDBC and ODBC.

For more on ADBC, refer to the [project](#) and [documentation](#), along with our other articles for benchmarks and examples of use:

- [Comparing Performance of ADBC and JDBC in Python for Arrow Flight SQL](#)
- [Explore a New Way to Deploy Data Storage and Analytics with Arrow Flight SQL and Apache Superset](#)

With that aside, time for the news.

An open source, incredibly powerful, and easily usable columnar database, DuckDB has quickly grown popular, having [over one million downloads a month on PyPI](#). Recently, [they added ADBC support](#) for clients who need columnar data transmission. As they point out in their article, JDBC and ODBC were developed in a row-based era, and using them created notable bottlenecks. With the advent of ADBC, there are performance gains to be found — as they said themselves, “[Due to DuckDB's zero-copy integration with the Arrow format, using ADBC as an interface is rather efficient, since there is only a small constant cost to transform DuckDB query results to the Arrow format.](#)”

This news is two-in-one, but Arrow all the way down. dbt, a system to develop, maintain, and deploy data transformation pipelines, has [updated its Semantic Layer software to a new beta version](#). This version has, “new APIs, including an entirely rebuilt JDBC interface built with ArrowFlight, allowing for more seamless integrations and applications to be built on top of the new Semantic Layer.” Arrow Flight is the wire protocol for Arrow columnar data, which ensures fast easy, and most importantly columnar data transfer. Alongside this development comes new [support for ADBC](#), enabling Arrow-native workflows via easy integration and performance with columnar data.

Finally, Snowflake, through a collaboration with Voltron Data, now supports ADBC! This enables columnar data exchange with systems and applications built in any language that can use ADBC — and, much like DuckDB, Snowflake's columnar nature ensures that there's no data conversion bottleneck as with JDBC or ODBC. As said in their [blog](#), “This makes it ideal for any bulk columnar analytics workflows, avoiding the cost of transposing data to a row-oriented format and back, making it much more efficient than ODBC/JDBC.”

The Future of Columnar Transmission with ADBC

Columnar data has become more popular due to its performance and total cost of ownership advantages in data analytics and data transfer, the missing piece being an easily integrated connector like JDBC and ODBC. ADBC solves this problem, and the industry has noticed. We look forward to more adoption. Every new use of ADBC makes columnar data even easier to use in production workloads and lets you integrate everything with ease.

Voltron Data helps companies integrate tools within the Arrow ecosystem like ADBC, Arrow Flight, nanoarrow, and more. To learn more about our approach, check out our [Product](#) page.