

The mountains Trino climbed

A project update for December 2023



Martin Traverso

By the numbers

Since Trino Fest in June

- **15** more releases, total of **29** in 2023
- **> 3000** additional commits, **> 5000** in 2023, **>35,000** total
- **685+** contributors
- **11,200+** Slack members

Continued growth and spread across industries, platforms,
and the globe.



What's happened since Trino Fest 2023?



General

- New support for SQL routines – details later
- Large performance improvements for aggregation queries
- More SQL improvements
 - Hex/binary/octal numeric literals
 - Unicode characters in string literals
 - FILTER on listagg function
 - SET SESSION AUTHORIZATION
- Improved schema evolution with nested fields



Object storage improvements

Continue improvements on the three pillars

- Performance
- Richer support for specific table format features
- Improved SQL support in connectors

Trino is a great foundation for lake and lakehouse use cases and data platforms.



Some improvement examples since Trino Fest

- Optimized Parquet reader/writer as default, removed old reader/writer
- New dedicated Avro reader and writer
- Performance improvements for ORC, Parquet, Text, and JSON
- Delta Lake
 - Support Databricks 13.3 LTS, Deletion Vectors, Writer version 7, \$properties
- Iceberg
 - Bearer token auth with Nessie



Object storage improvements

```
CALL example.system.register_table(  
  schema_name => 'testdb',  
  table_name => 'customer_orders',  
  table_location => 's3://my-bucket/a/path');  
  
SELECT * FROM "test_table$properties";  
  
CREATE OR REPLACE TABLE ...  
  
DROP SCHEMA lake.testdata CASCADE;  
  
TRUNCATE TABLE lake.testdata.visits;  
  
ALTER TABLE lake.testdata.visits DROP COLUMN ip;
```



Various other connector improvements

- Various: support for UPDATE
- Bigquery: support for DELETE
- Google Sheets: support for INSERT
- MariaDB: support for table statistics
- MongoDB: support for ALTER TABLE ... RENAME COLUMN
- Various: support for CASCADE option in DROP SCHEMA



SQL routines in Trino

Another important milestone

- Statically defined, built-in functions only ✓
- Dynamically resolved from plugin/connector ✓
- Dynamically defined with SQL language - SQL routine now ✓
- Dynamically defined with other language - future



Inline routines

Define and use within one query

```
WITH FUNCTION dots(input varchar)
RETURNS varchar
BEGIN
  IF length(input) > 15 THEN
    RETURN substring(input, 1, 12) || '...';
  END IF;
  RETURN input;
END
SELECT
  dots('A long string that will be shortened'),
  dots('A short string');
```



Catalog routines

- Define once and store in schema of catalog
- Use many times by all users

```
CREATE FUNCTION dots(input varchar)
RETURNS varchar
BEGIN
  IF length(input) > 15 THEN
    RETURN substring(input, 1, 12) || '...';
  END IF;
  RETURN input;
END;

SELECT dots('A long string that will be shortened');

SELECT dots('A short string');

SHOW FUNCTIONS FROM brain.default;

DROP FUNCTION dots(varchar);
```



Experimental support for dynamic catalogs

- Functional feature, documentation available
 - CREATE CATALOG example USING memory;
- Secrets management and protection pending
- Usability improvements pending
 - No SHOW CREATE CATALOG yet
- Integration with different deployment models pending
- Looking for feedback



Other features and news

- Breaking changes highlighted in release notes
 - For example: `• ⚠️ Remove support for late materialization,`
- Numerous Helm chart improvements:
 - Sidecar containers, secrets, secret mounts, common labels, coordinator ingress
- New [TypeScript client library](#) from the community
- New [Trino Gateway](#) project – more in following session



Roadmap

- Snowflake connector close to shipping
- Object storage caching with Alluxio getting close
- `json_table` function
- Continued Project Hummingbird performance improvements
- Testing refactor and CI improvements initiative



Java 21

- Trino now supports Java 21
 - Chosen runtime for Docker container
 - Java 17 still supported
- Java 21 will be required ***really soon***
- Help us test!

Let's talk about benefits and next steps



Trino community

- [Events and meetups](#) in India, Japan, South Korea, USA, and others
- Trino Community Broadcast, up to episode 53
- [Rich ecosystem overview](#) on the website
- [SQL training series](#) completed 4 episodes
- Trino Contributor Congregation coming up in early 2024



Questions?



SQL training series – move

- Four episodes so far with over four hours of demos
- Lots of SQL queries and presentations available for you all
- Over 100 live attendees live each time, many more views
- Find the videos, slide decks, and sample scripts on the Trino blog



Let us know what you want to learn next,
and stay tuned for more classes.

Trino Contributor Congregation

- Coming early 2024
- Enable discussion and help for numerous topics:
 - Kubernetes operator and Helm chart future
 - Trino Gateway next steps
 - Java 21 feedback
 - Project Hummingbird update
 - Impact and help with TrinoFileSystem refactor
 - Current pull requests
 - Plans for future contributions



Five reasons for attending Trino Summit 2023

1. Get latest and greatest Trino news
2. Learn about real-world Trino experience
3. Meet your peers and learn together
4. Have fun with Trino community
5. Attend from anywhere



Aggregated release notes



Highlights from release notes

Highlights

General

- * Add support for the `{func}`any_value`` aggregation function. ([{issue}`17777`](#))
- * Add support for underscores in numeric literals. ([{issue}`17776`](#))
- * Add support for hexadecimal, binary, and octal numeric literals. ([{issue}`17776`](#))
- * Add support for check constraints in an ``UPDATE`` statement. ([{issue}`17195`](#))
- * Add support for adding nested fields with an ``ADD COLUMN`` statement. ([{issue}`16248`](#))
- * Add support for renaming nested fields in a column via ``RENAME COLUMN``. ([{issue}`16757`](#))
- * Add support for setting the type of a nested field in a column via ``SET DATA TYPE``. ([{issue}`16959`](#))
- * Add support for comments on materialized view columns. ([{issue}`18016`](#))
- * Add support for displaying all Unicode characters in string literals. ([{issue}`5061`](#))
- * Add support for ``SET SESSION AUTHORIZATION`` and ``RESET SESSION AUTHORIZATION``. ([{issue}`16067`](#))
- * Add support for comparing IPv4 and IPv6 addresses and CIDRs with `[contains](ip-address-contains)`. ([{issue}`18497`](#))
- * Improve performance of ``GROUP BY`` and ``DISTINCT``. ([{issue}`19059`](#))
- * Allow `{doc}`/sql/show-functions`` for a specific schema. ([{issue}`19243`](#))
- * Add security for function listing. ([{issue}`19243`](#))
- * Add support for `[](/routines)`. ([{issue}`19308`](#))
- * Add support for `[](/sql/create-function)` and `[](/sql/drop-function)` statements. ([{issue}`19308`](#))
- * Add support for the ``REPLACE`` modifier to the ``CREATE TABLE`` statement. ([{issue}`13180`](#))
- * Add support for a ``FILTER`` clause to the ``LISTAGG`` function. ([{issue}`19869`](#))

Docker image

- * Update Java runtime to Java 21. ([{issue}`19553`](#))

Security

- * Add support for configuring authorization rules for ``ALTER ... SET AUTHORIZATION...`` statements in file-based access control. ([{issue}`16691`](#))
- * Add support for validating JWT types with OAuth 2.0 authentication. ([{issue}`17640`](#))
- * Support defining rules for procedures in file-based access control. ([{issue}`19416`](#))

JDBC driver