Presto Summit NYC 2019

Martin Traverso, Dain Sundstrom, David Phillips

Presto Software Foundation

“An independent, non-profit organization with the mission of supporting a community of passionate users and developers devoted to the advancement of the Presto distributed SQL query engine for big data.”

“It is dedicated to preserving the vision of high quality, performant, and dependable software.”

“Ensuring the project remains open, collaborative and independent for decades to come”

By the Numbers

- Presto Software Foundation launched on January 31, 2019
- 27 releases (1-2 weeks between releases)
- 2700+ commits
    - 200k lines changed (> 20% of the codebase)
- 1500+ pull requests closed
- 100+ contributors
- 280+ weekly active members on Slack

Improvements

- FETCH FIRST … WITH TIES syntax
- OFFSET syntax
- COMMENT ON <table> IS …
- [LEFT/RIGHT/FULL] JOIN LATERAL (…) ON
- IGNORE NULLS for window functions
- .* for ROW expressions
- Pass-through security (client provided credentials)
- Impersonation for Hive Metastore
- Kerberos security improvements
- Support for Hadoop KMS
- Role-based security
- Secure query results in client API
- Current user security mode for views
- Support for Azure Data Lake
- Hive Bucketing V2
- Docker image
- Spill-to-disk improvements
- CLI output formats
- Syntax highlighting in CLI
- UUID type and functions
- format(), combinations() functions
- ORC bloom filters (non-legacy)
- Connector-provided view definitions
- Elasticsearch Connector
- Google Sheets Connector
- Amazon Kinesis Connector
- Apache Phoenix Connector
- LZ4/ZSTD support for ORC/Parquet
- More type mappings for various connectors
- Performance improvements for GCS and S3
- Performance improvements for UNNEST

… and more! https://prestosql.io/docs/current/release.html

Contributors

MichaelChirico
ajorgens
linxingyuan1102
asrivastav
sshardool
bill-warshaw
apc999
hustnn
wagnermarkd
Lewuathe
vkorukanti
luohao
jlabarbera11
mattsfuller
aalbu
MiguelWeezardo
pgagnon
dain
bryanck
BenoitHanotte
guerreromdq
amoghmargoor
xumingming
qqibrow
JamesRTaylor
martint
Praveen2112
guyco33
sopel39
zhenxiao
kranthikiran01
amiorin
kokosing
raunaqmorarka
11xor6
electrum
chancez
cdw9bf
ptkool
MarvinCai
vincentpoon
yui-knk
rzeyde-varada
anoopj
ebyhr
dpolonsky
kasiafi
stagraqubole
takezoe
mosabua
Yaliang
elonazoulay
wyukawa
ankitdixit
VicoWu
eskabetxe
findepi
jvanzyl
dilipkasana
garvit-gupta
anusudarsan
ryanrupp
Anurag870
ChethanUK
ilfrin
pettyjamesm
kabunchi
jirassimok

aws

Appian

Gett

hulu

LINE

lyft

Qu

salesforce

Uber

Alibaba Group
阿里巴巴集团

VARADA

ZOHO

Presto Community

- Github: https://github.com/prestosql
- Website: https://prestosql.io
- Blog: https://prestosql.io/blog
- Twitter: @prestosql
- Slack: prestosql.slack.com

Roadmap

Roadmap

- Dynamic filtering
- Row dereference pushdown
- Iceberg connector
- Pinot connector
- Function revamp

Presto Functions

Function Plugins

- Plugin `Set<Class<?>> getFunctions()`
- Class annotations determine function signature
- All functions share a single global namespace
- Functions registered when Plugin is loaded (e.g., startup)

Example Function

```
@Description("Returns a randomly generated UUID")
@ScalarFunction(value = "uuid", deterministic = false)
@SqlType(StandardTypes.VARCHAR)
public static Slice UUID()
{
    return Slices.utf8Slice(UUID.randomUUID().toString());
}
```

Problems

- All functions in single namespace
- All functions registered up front
- Function can not be updated
- Only annotated Java classes supported
- Not configurable

Namespaces

- SQL specification attaches functions to a schema
- Functions can be referenced by an absolute name:

```
catalog.schema.function
```

- There is a search path for non-absolute function names
- The SET PATH statement can be use to change this path

Connector Functions

- Existing connector API manages a catalog
- Extend this API to resolve functions dynamically

```
getFunctions(Name)::Collection<FunctionMetadata>

getImplementation(FunctionId, ActualSignature)::Function
```

- Part of existing transaction framework

SQL Defined Functions

- Create/Drop function
- Query "local" functions?
- SQL procedure language (BASIC like)
- Supports simple single expression and procedural blocks

WITH FUNCTION

```
WITH FUNCTION hello(s VARCHAR)
RETURNS VARCHAR
RETURN 'Hello ' || s || '!'
SELECT hello('world')
```

```
WITH FUNCTION times100(a INT)
RETURNS INT
BEGIN
    DECLARE x INT DEFAULT cast(100 as INT);
    RETURN x * a;
END
SELECT times100(42)
```

Language Interface

- SQL allows many languages for functions
- Connector responsible for storing "definition":

```
createFunction(FunctionMetadata, lang, definition)

getDefinition(FunctionId)::definition
```

- Presto will suport a text to method transformer:

```
compileFunction(definition)::MethodHandle
```

Remote Functions

- Support function which cannot or should not run in the same process as Presto
  - Untrusted or insecurity
  - Unstable or unreliable
  - Resource intensive
  - Connect to functionality in remote systems

Batch Calling Convention

- Batch invocations for efficiency
- Interface will be something like:

```
myFunction(Page arguments)::Block
```

- Requires changes to isolate remote functions in complex expressions

Polymorphic Table Functions

- Added in SQL 2016
- Table function produces a collection of rows (e.g., a table)
  - UNNEST is a table function defined in the spec
- Table function has a predefined (static) signature
- Polymorphic table function has a signature that is derived dynamically based on the function arguments
  - Powerful enough to define virtually all SQL features

Example

```
SELECT *
FROM TABLE (
  CSVreader(
    file => 'abc.csv',
    floats => DESCRIPTOR ("principle", "interest")
    dates => DESCRIPTOR ("due_date")))
```

```
SELECT D.Region, R.Name, R.Value
FROM TABLE (
  ExecScript(
    script => '...',
    input => TABLE (SELECT foo, bar FROM t) AS D
    rowtype => DESCRIPTOR (name VARCHAR(100), value REAL)))
```

Getting Involved

- Join Slack
  - https://prestosql.io/community.html
  - #troubleshooting channel
  - #dev channel
- File issues/bugs:
  - https://github.com/prestosql/presto
- Write blog posts
  - https://prestosql.io/blog