

Migrating to Hive 2020.1

Treasure Data is migrating to a new version of the Hive Engine: Hive 2020.1, also referred to as Hive 2/Tez. Treasure Data is migrating because Hive 2020.1 complies with ANSI SQL standards and allows Treasure Data to utilize future release enhancements.

The following outline summarizes a possible Hive0.13 to Hive 2020.1 migration path. The goals are:

- Minimize disruption to workloads currently running successfully in Hive 0.13.
- Maximize the benefits from Hive 2 in stages appropriate to the business.

Migration Stages

Testing

Treasure Data recommends you run queries with Hive 2020.1 for testing purposes:

- **Modify Existing Queries:** Update your existing queries to remove Hive 2 incompatible syntax. We recommend rewriting your queries for stricter Hive 2020.1-compatible syntax and then run the modified queries in Hive 0.13. For more information on syntaxes, see [Hive 0.13](#) and [Hive 2020.1](#).
- **New Queries:** Make sure they are compatible with Hive 2020.1 syntax and avoid the incompatible syntax from Hive 0.13.
- **Test Workloads:** Work with your technical account management or support to test new and existing workloads for use with Hive 2020.1.

Production

Treasure Data suggests that you use Hive 2020.1 for the following workload types:

- Production workloads
- New ad-hoc and batch workloads

You should also review and upgrade all other existing workloads.

Default Processing Engine

- **TD Console:** Hive users will see Hive 2020.1 as the default Hive version.
- **CLI / API:** If no version is specified, Hive 2/Tez is used after changing the default processing engine.

Run Queries

Review the following table to understand how to run Hive 0.13 and Hive 2/Tez queries.

Hive Engine	Run Hive 0.13 Query	Run Hive 2/Tez Query
Default = Hive 0.13	No changes	Override the engine version from <i>None</i> to <i>Stable</i> .
Default = Hive 2020.1	Explicitly specify engine_version='0.13'	No changes

Semantic Checking Process

There are a couple of options available for the semantic checking of Hive queries; both positive and negative impacts are explained.

Test Query

Run 'EXPLAIN <query>'

Positive Impact

No impact on production
Check SQL compliance easily

Negative Impact

The statement prints the query plan only; you cannot look at the result.

Run Query / Insert into Temporary Table

Positive Impact

Make comparisons on results.

Negative Impact

If you do not rewrite insert queries, Treasure Data will insert the results into production tables for testing.

EXPLAIN Example

The EXPLAIN command can help check for errors without having to execute a query.

```
1 explain select count (*) from call center;
```

The following is a sample of a result from the call:

```
1 FAILED: ParseException line 1:29 cannot recognize input near 'from' 'call center'<EOF> in join source
```

See also:

- [Hive 0.13 and Hive 2 Code and Syntax Compatibility](#)
- [Hive Reserved and Non-Reserved Keywords](#)