

Speed up Queries that Produce Massive Results

The following SQL queries might help you speed up your query results.

This topic includes:

- [Use CREATE TABLE AS or INSERT INTO](#)
- [Use CREATE TABLE \(table\) AS SELECT ...](#)
- [Use INSERT INTO \(table\) SELECT ...](#)
- [Improve Very Large Query Performance](#)
- [Avoid using SELECT *](#)
- [Use result_output_redirect](#)

Use CREATE TABLE AS or INSERT INTO

```
SELECT * FROM my_table
```

Presto uses JSON text to materialize query results. If the preceding table contains 100GB of data, the coordinator transfers more than 100GB of JSON text to save the query result.

Use CREATE TABLE (table) AS SELECT ...

```
DROP TABLE IF EXISTS my_result;  
CREATE TABLE my_result AS SELECT * FROM my_table;
```

You can parallelize the query result output process by using CREATE TABLE AS SELECT statement. To clean up the result table beforehand, add a DROP TABLE statement at the top of your query. The result output performance will be 5x ~ faster than running SELECT *. Our Presto skips the JSON output process and directly produces a 1-hour partitioned table.

Use INSERT INTO (table) SELECT ...

You can also use INSERT INTO (table) AS SELECT ... to append the data to an existing table. It can improve the query result output performance:

```
CREATE TABLE IF NOT EXISTS my_result(time bigint);  
INSERT INTO my_result SELECT * FROM my_table;
```

If the subsequent SELECT statement does not produce the time column, INSERT INTO attaches the query execution time as *time* column values. You can find the inserted rows using the query execution time.

Improve Very Large Query Performance

You can parallelize the query result output process by using the CREATE TABLE AS SELECT statement. If you DROP the table before running the query, your performance is significantly better. The result output performance will be 5x faster than running SELECT *. Treasure Data Presto skips the JSON output process and directly produces a 1-hour partitioned table.

Without using DROP TABLE, Presto uses JSON text to materialize query results. And if the result table contains 100GB of data, the coordinator transfers more than 100GB of JSON text to save the query result. So, even if the query computation is almost finished, the output of the JSON results takes a long time.

To clean up the result table beforehand:

1. Add a DROP TABLE statement at the top of your query.
2. Use CREATE TABLE (table) AS SELECT ...

For example, your query might look like the following.

```
DROP TABLE IF EXISTS my_result;  
CREATE TABLE my_result AS  
  SELECT * FROM my_table;
```

Avoid using SELECT *

```
SELECT * FROM my_table
```

Treasure Data uses a column-oriented storage format, so accessing a small set of columns is really fast, but as the number of columns increases in your query, it deteriorates the query performance. Be selective in choosing columns.

```
SELECT id, name, address FROM my_table
```

Use result_output_redirect

To improve the job performance of queries that produce massive results, *result_output_redirect* is used to enhance the performance of Presto jobs. By default, *result_output_redirect* is set to 'true'.

This impacts performance even though the query results are ready; TD Presto is waiting for the worker node to complete its sequential operations. To mitigate this, Treasure Data now uses *result_output_direct*, which redirects the query result to S3 in parallel, thereby improving the performance of the queries.

LIMITATION: There are SQL queries where Hints do not improve performance, including:

- When a query includes ORDER BY
- When a query is CREATE TABLE or INSERT TABLE
- When a query includes uppercase letters
- When a query has duplicated column names