**EXPLAIN and EXPLAIN ANALYZE**

In Starburst (and Trino, which Starburst is based on), the EXPLAIN and EXPLAIN ANALYZE commands are used to understand and analyze how a query will execute, which is critical for optimizing performance, especially in distributed data environments.

**1. EXPLAIN**

The EXPLAIN command shows the query execution plan without actually running the query. This plan is a step-by-step outline of how Starburst will process the query. It breaks down each phase, from data access to the various joins, aggregations, and final results.

**Key Features**:

* **Logical Plan**: Provides a breakdown of how the query is interpreted logically, including each step and operation, without involving physical data processing.
* **Stages and Distribution**: Shows how data processing will be distributed across nodes, which is crucial in understanding parallelism and resource usage.
* **Joins and Aggregations**: Displays the type of joins (e.g., broadcast, partitioned) and aggregations that will occur.

**Use Cases**:

* **Query Optimization**: Helps identify potential inefficiencies, such as unnecessary full table scans or inappropriate join types.
* **Understanding Data Distribution**: Assists in verifying that large tables are not being moved unnecessarily, which can slow down performance.

**Example**:

EXPLAIN SELECT \* FROM sales WHERE total > 1000;

**2. EXPLAIN ANALYZE**

The EXPLAIN ANALYZE command goes a step further by actually executing the query and providing real runtime statistics, in addition to the logical execution plan. It’s essentially a profiling tool, showing the actual time taken and resources used at each stage of the query.

**Key Features**:

* **Execution Time**: Reports the actual execution time of each stage, helping pinpoint which parts of the query are slow.
* **Memory and CPU Usage**: Provides insight into resource consumption at each step, allowing you to diagnose memory bottlenecks or CPU-heavy operations.
* **Distributed Performance**: Shows performance metrics across distributed nodes, which helps identify data skew or inefficient data distribution.

**Use Cases**:

* **Diagnosing Performance Issues**: Helps you see where time is spent in the execution, such as long-running joins, aggregations, or data shuffling.
* **Resource Usage Analysis**: Lets you detect stages that may be resource-intensive, helping with resource allocation and workload balancing.
* **Comparing Optimizations**: Use it to test optimizations in real-time, by comparing metrics before and after query adjustments.

**Example**:

EXPLAIN ANALYZE SELECT \* FROM sales WHERE total > 1000;

**Key Differences Between EXPLAIN and EXPLAIN ANALYZE**

| **Feature** | **EXPLAIN** | **EXPLAIN ANALYZE** |
| --- | --- | --- |
| **Execution** | Logical plan only (no data processed) | Executes the query and provides real runtime data |
| **Resource Usage** | No actual resource measurement | Reports actual memory, CPU usage, and time spent |
| **Use Cases** | Optimizing query structure | Optimizing query performance with real metrics |
| **Performance Impact** | Low, as no data is processed | Higher, as it executes the query in full |

**Practical Applications**

* **Initial Tuning with EXPLAIN**: Start with EXPLAIN to assess the logical plan, especially for larger queries, so you can avoid actual execution if changes are needed.
* **Detailed Analysis with EXPLAIN ANALYZE**: Once the logical structure is optimized, use EXPLAIN ANALYZE for detailed diagnostics and to finalize optimizations with real-time feedback.

Both commands are critical for Starburst query optimization, helping users and developers fine-tune performance in distributed and large-scale data environments.