

# Assignment -10

## 1. Analyze and Improve the Performance of a Sample Dataset

Steps:

Profile queries on a given dataset using the Query Profiler.

Optimize the queries by:

Adding filters and clustering keys.

Avoiding redundant joins.

Document the improvements with before-and-after execution metrics.

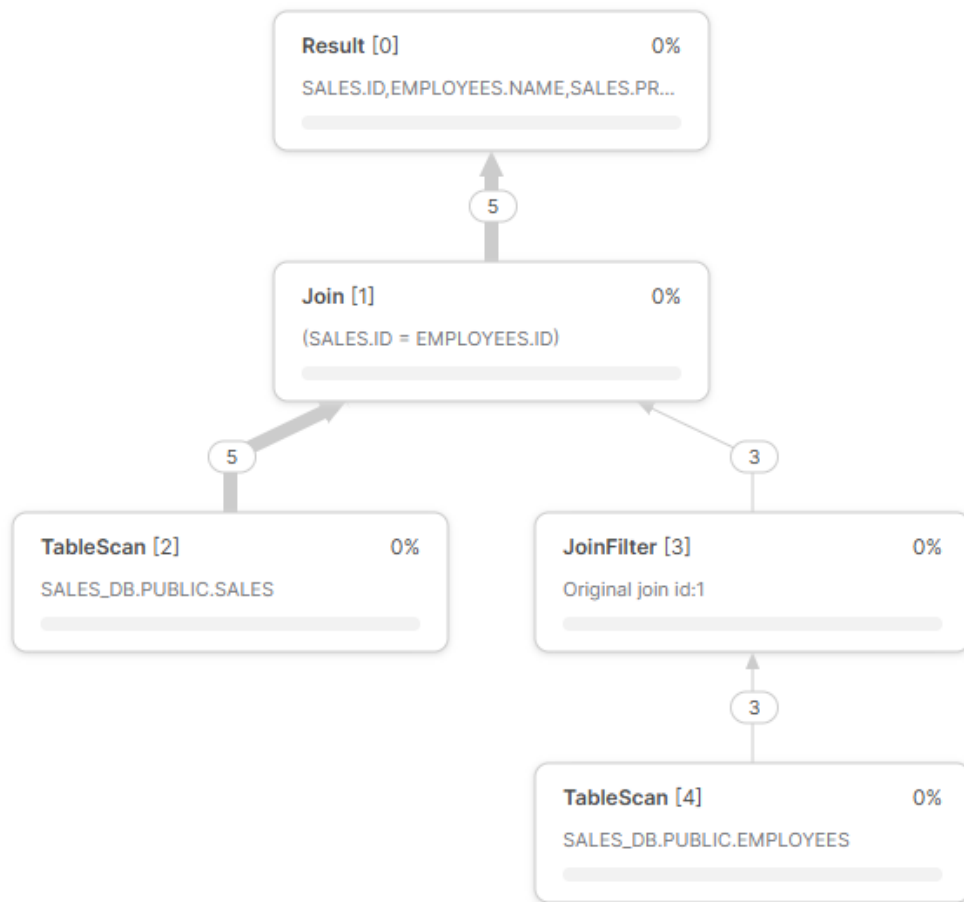
```
SELECT sales.ID, employees.name, sales.PRODUCT_NAME, employees.SSN  
FROM SALES_DB.PUBLIC.SALES  
JOIN SALES_DB.PUBLIC.EMPLOYEES ON sales.ID = employees.ID;
```

### Profile Overview (Finished)

Total Execution Time	(33ms) 100.0%
• Initialization	100.0%

### Statistics

Scan progress	100.00%
Bytes scanned	0.00MB
Percentage scanned from cache	100.00%
Partitions scanned	2
Partitions total	2



Now when we apply filter in the same query obviously it will take more time for execution than the previous one:

```
SELECT sales.ID,employees.name, sales.PRODUCT_NAME, employees.SSN
FROM SALES_DB.PUBLIC.SALES
JOIN SALES_DB.PUBLIC.EMPLOYEES ON sales.ID = employees.ID
WHERE QUANTITY>10;
```



### Profile Overview (Finished)



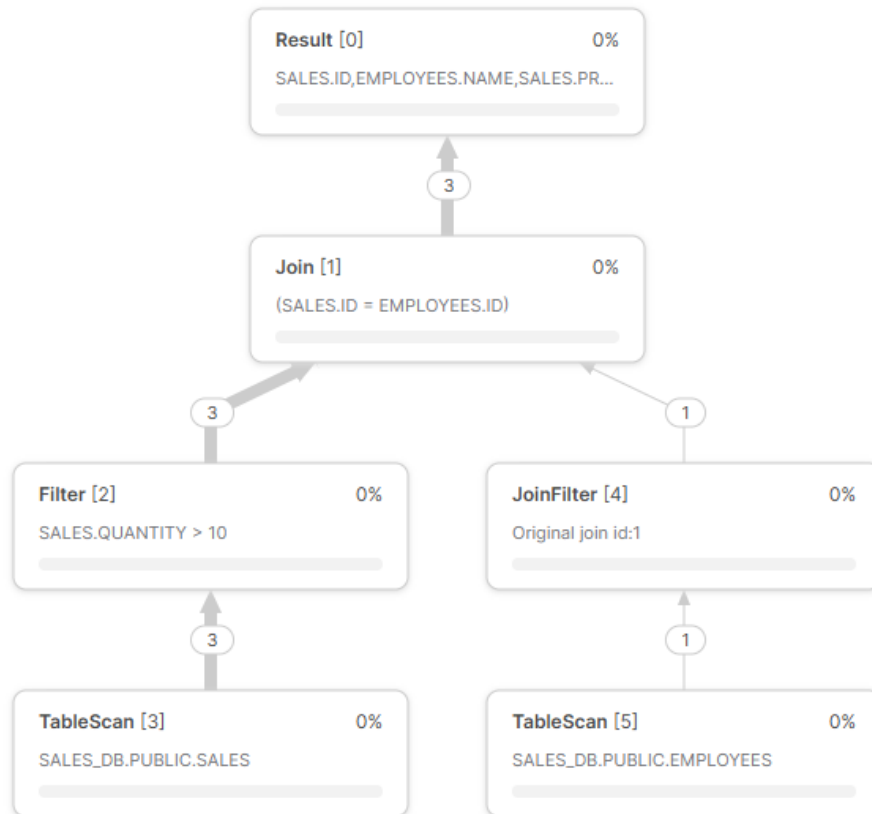
Total Execution Time	(452ms) 100.0%
Remote Disk I/O	33.3%
Initialization	66.7%

### Statistics

Scan progress	100.00%
Bytes scanned	0.00MB
Percentage scanned from cache	0.00%
Partitions scanned	2
Partitions total	2

After applying clustering, there is drastic decrease in execution time for the same filtering query above:

```
ALTER TABLE SALES CLUSTER BY (QUANTITY);
```



### Profile Overview (Finished)

Total Execution Time (33ms) 100.0%

- Initialization 100.0%

### Statistics

Scan progress	100.00%
Bytes scanned	0.00MB
Percentage scanned from cache	100.00%
Partitions scanned	2
Partitions total	2

## 2. Write a Cost Management Strategy for Snowflake Usage

- Best practices for warehouse management.

### Use Appropriately Sized Warehouses:

Example: Query runtime on Small: 4 mins

Same query on Medium: 1 min

Cost:

Small (1 credit/hour): 4 mins → 0.067 credits

Medium (2 credits/hour): 1 min → 0.033 credits

Medium is faster and cheaper.

### Enable Auto-Suspend and Auto-Resume:

It prevents idle warehouses from consuming credits.

- Methods to monitor and reduce query costs.

### Query Profiling:

Use Query History + Profile tab to identify high-cost queries using Query ID, Execution Time, and Bytes Scanned.

### Use Caching Effectively:

Cached results = 0 compute cost.

Reuse query results when possible by avoiding unnecessary refreshes.

### Add filters and apply clustering

- Strategies for scaling virtual warehouses based on workload.
- Include examples and calculations based on sample query metrics.

### Vertical Scaling (Resizing the Warehouse):

Scenario: You have a warehouse running queries that typically take 5 minutes to complete on a medium warehouse.

Action: During peak hours, you notice the queries are taking longer due to increased load. You resize the warehouse to large.

Calculation: If a medium warehouse costs 2 credits per hour and a large warehouse costs 4 credits per hour, running the large warehouse for 1 hour during peak times will cost 4 credits instead of 2.

### Horizontal Scaling (Adding Clusters):

Horizontal scaling involves adding more clusters to a multi-cluster warehouse, which is useful for handling concurrent queries.

Scenario: Your warehouse handles 100 concurrent queries during peak hours.

Action: You configure the warehouse to scale out to 3 clusters during peak times.

Calculation: If each cluster of a medium warehouse costs 2 credits per hour, running 3 clusters will cost 6 credits per hour

### Separate Warehouses for Different Workloads:

Create separate warehouses for different types of workloads, such as data loading and query execution, to optimize performance and cost.