Kannika Armstrong
TCSS 564 Database Systems Internals
April 21, 2025

Database Systems Internals - Project
Query Optimization
Part 1: query plan change in response to changes in the DB schema

1. Database Information

- ***My database name:*** "WeatherDB" with 4 tables: Stations, DailyWeather, WindConditions, and WeatherAlerts
- No Primary key, Foreign key, and indexes
- All data are unsorted in every table

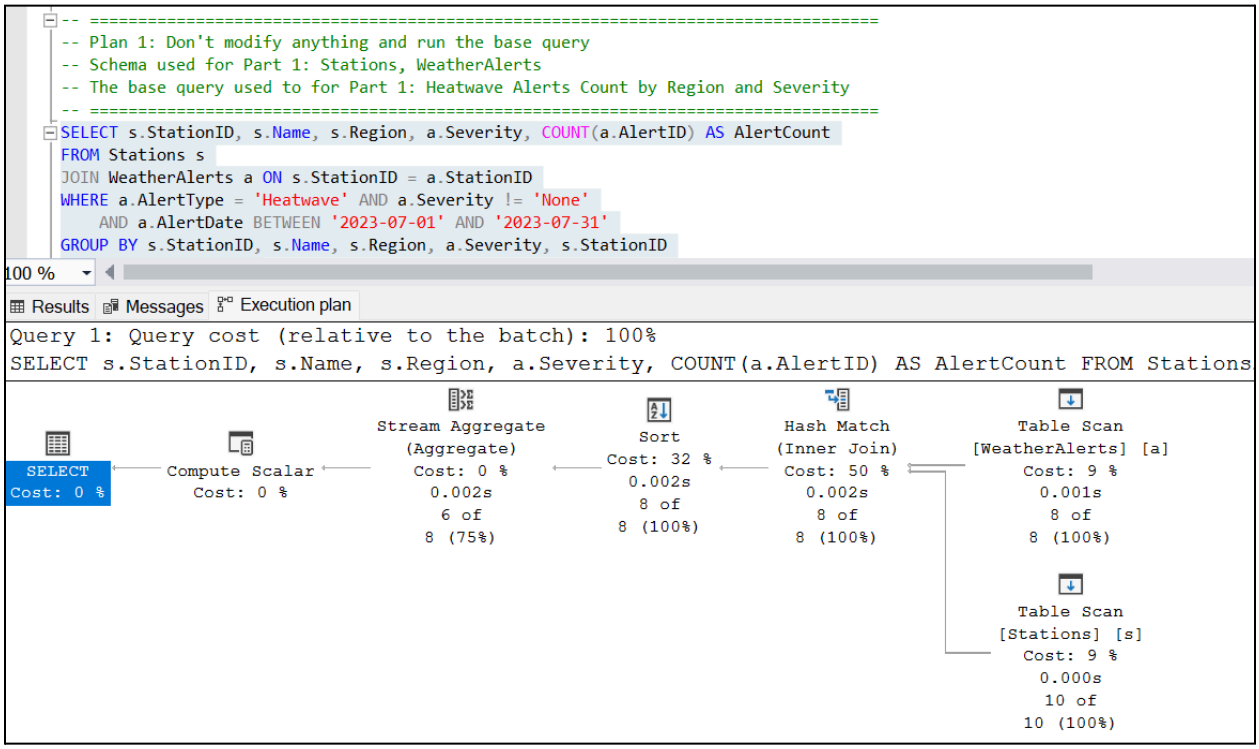
Table Name	Column Names	Number of Rows
Stations	StationID, Name, Region, Latitude, and Longitude	10
DailyWeather	WeatherID, StationID, WeatherDate, MaxTempC, MinTempC, AvgTempC, PrecipitationMM, and HumidityAvg	65
WindConditions	WindID, StationID, WindDate, WindSpeedKMH, and WindDir	65
WeatherAlerts	AlertID, StationID, AlertDate, AlertType, and Severity	65

2. The base query used for Part 1:

```
SELECT
s.StationID, s.Name, s.Region, a.Severity, COUNT(a.AlertID) AS AlertCount
FROM Stations s
JOIN WeatherAlerts a ON s.StationID = a.StationID
WHERE a.AlertType = 'Heatwave' AND a.Severity != 'None'
      AND a.AlertDate BETWEEN '2023-07-01' AND '2023-07-31'
GROUP BY s.StationID, s.Name, s.Region, a.Severity
ORDER BY s.StationID;
```

Query Plan 1: Don't modify anything and run the base query

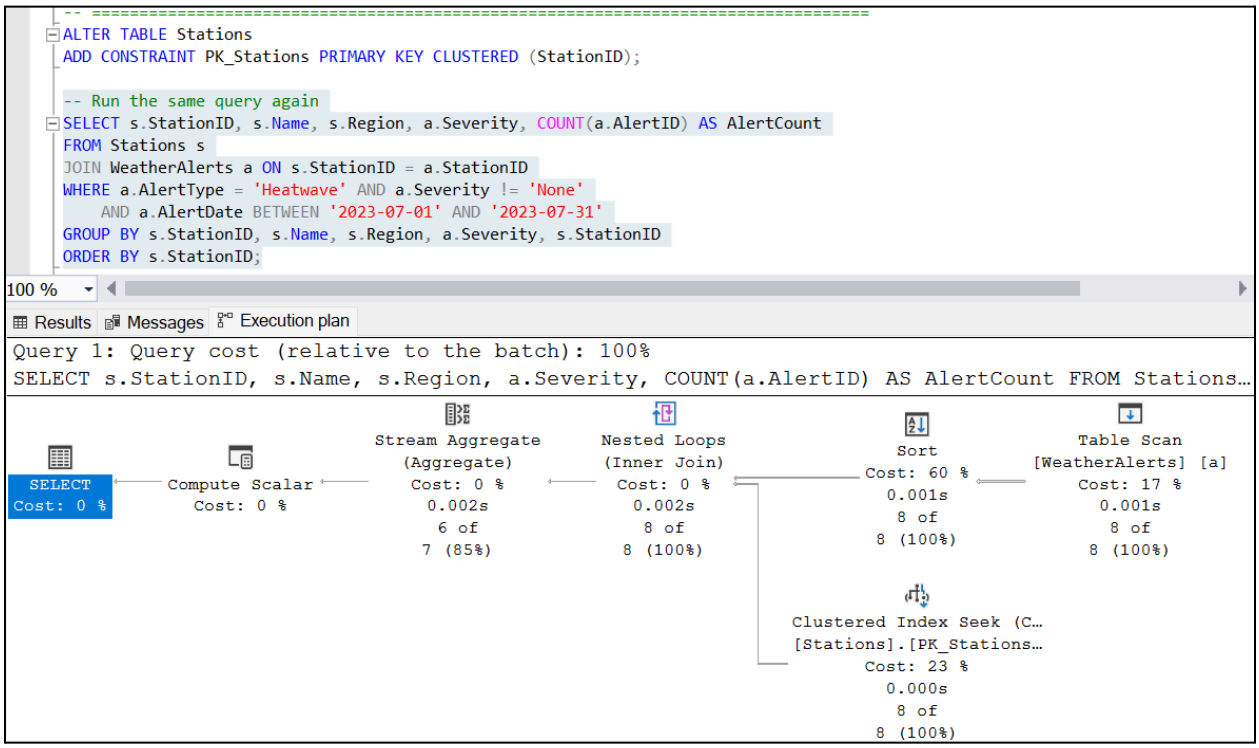
1. ***Screenshot of the query plan***



2. ***Changes made in the schema:***
   Run the base query without creating/dropping indexes, adding constraints, and changing the index types
3. ***Optimizer behavior:***
   Since none of the tables have a Primary Key, Foreign Keys, Clustered Indexes, or Non-Clustered Indexes, the query optimizer chooses to use:
   - The Table Scan on both `'WeatherAlerts'` and `'Stations'` tables
   - Build the hash table on `'WeatherAlerts'`, then use Hash Match (Inner Join) for joining both input datasets from `'WeatherAlerts'` and `'Stations'` tables
   - After joining the tables, the query optimizer chooses to sort the data, then uses the `'Stream Aggregate'` for the `'AlertCount'`
4. ***Cost and time:***
   - `Estimated Subtree Cost`: 0.0360782
   - `CPU Time` (stats): 0 ms
   - `Elapsed Time` (stats): 97 ms
   - `Actual Execution Time` (Elapsed Time): 613 ms

Query Plan 2: Adding Primary Key Clustered Index in the Stations table

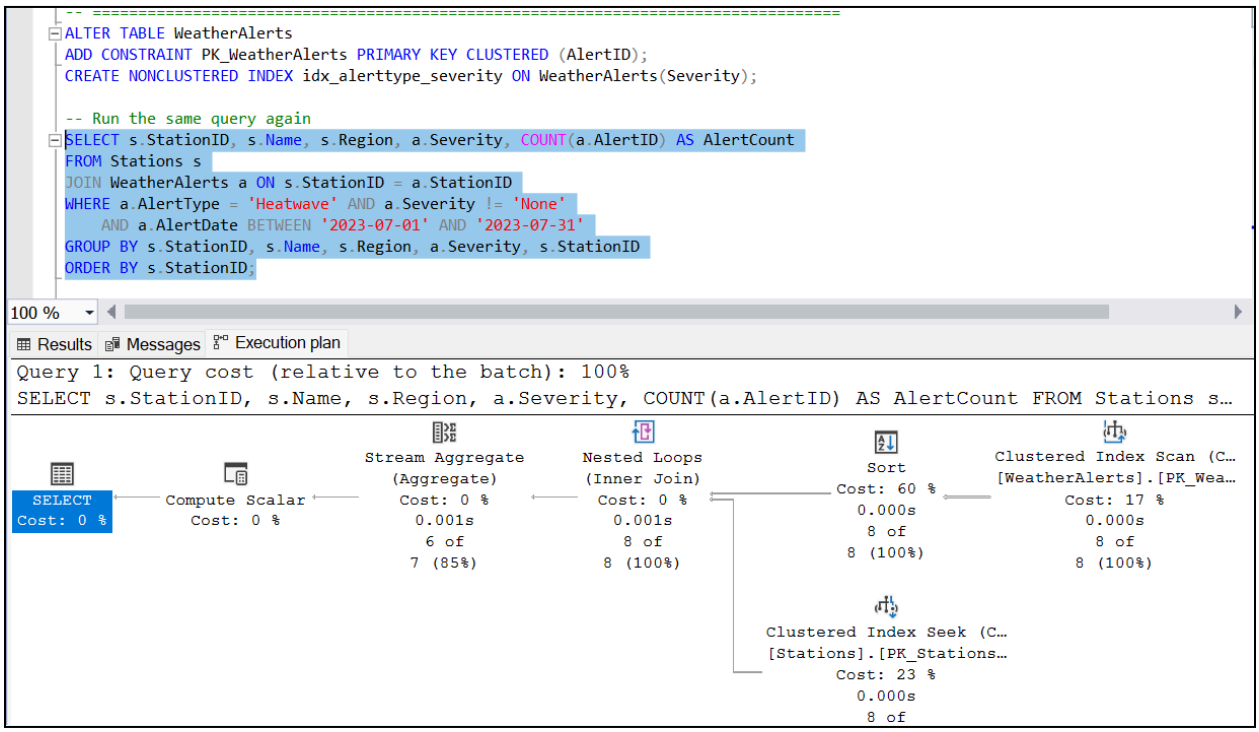
1. ***Screenshot of the query plan***



2. ***Changes made in the schema:***
   Adding a Primary Key Clustered Index on Stations(StationID) to help with the ORDER BY clause, then run the base query
3. ***Optimizer behavior:***
   Now, there is a Primary Key Clustered Index on Stations(StationID), the query optimizer chooses to use:
   - The Clustered Index Seek on the `'Stations'` table, but the Table Scan and Sort on the `'WeatherAlerts'` table
   - Nested Loop (Inner Join) to join those 2 tables together because the dataset has only 8 rows, so the query optimizer chooses to use the Nested Loop instead of Merge Join, even though those 2 datasets are already sorted by StationID
   - The datasets after joining are already sorted, so the query optimizer chooses to use `'Stream Aggregate'` for the `'AlertCount'`
4. ***Cost and time:***
   - `Estimated Subtree Cost`: 0.0192801
   - `CPU Time` (stats): 0 ms
   - `Elapsed Time` (stats): 25 ms
   - `Actual Execution Time` (Elapsed Time): 125 ms

Query Plan 3: Adding Primary Key Clustered Index in the WeatherAlerts table, creating a Non-Clustered Index on the Severity

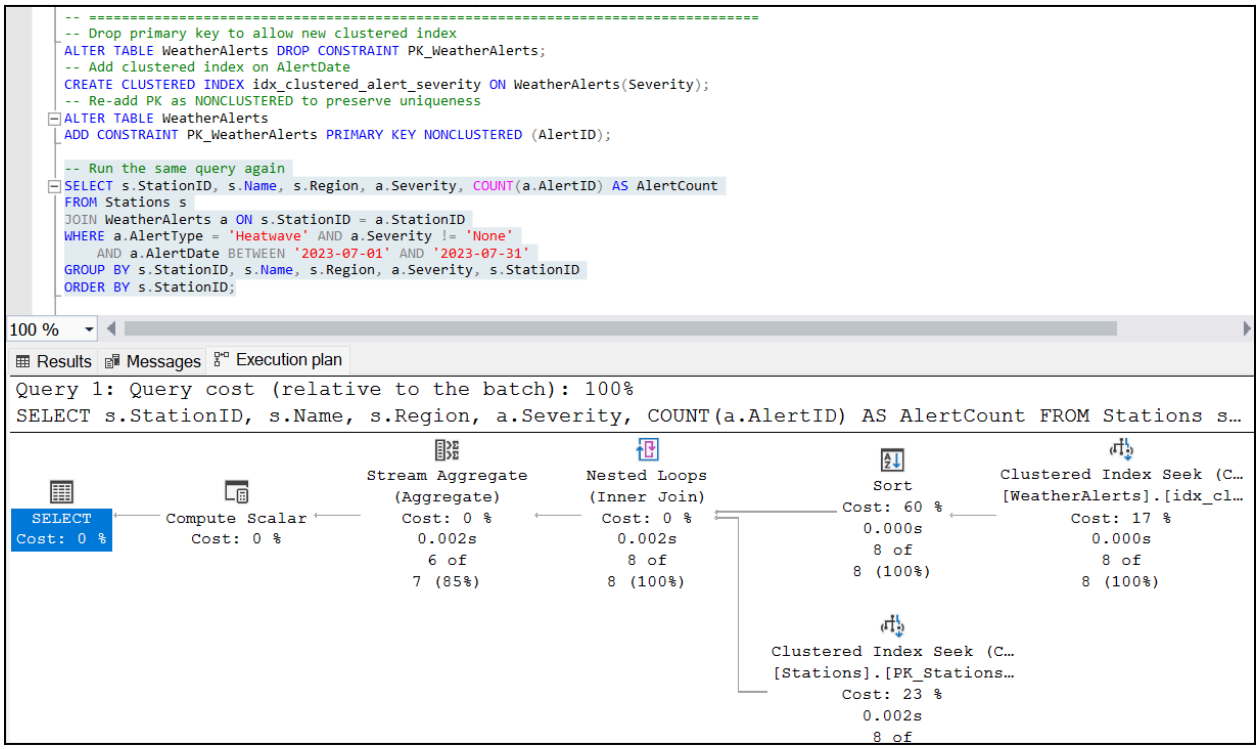
1. ***Screenshot of the query plan***



2. ***Changes made in the schema:***
   Adding Primary Key Clustered Index on WeatherAlerts(AlertID) and creating Non-Clustered Index on WeatherAlerts(Severity)
3. ***Optimizer behavior:***
   Now, there is a Primary Key Clustered Index on Stations(StationID), a Primary Key Clustered Index on WeatherAlerts(AlertID), and a Non-Clustered Index on WeatherAlerts(Severity), the query optimizer chooses to use:
   - The Clustered Index Scan on the `'WeatherAlerts'` table instead of the Table Scan, then Sort the data
   - Keep using Clustered Index Seek on the `'Stations'` table as Plan 2
   - Nested Loop (Inner Join) to join those 2 tables together because the dataset has only 8 rows, so the query optimizer chooses to use the Nested Loop instead of Merge Join, even though those 2 datasets are already sorted by StationID
   - The datasets after joining are already sorted, so the query optimizer chooses to use `'Stream Aggregate'` for the `'AlertCount'`
4. ***Cost and time:***
   - `Estimated Subtree Cost`: 0.0192801
   - `CPU Time` (stats): 0 ms
   - `Elapsed Time` (stats): 22 ms
   - `Actual Execution Time` (Elapsed Time): 94 ms

Query Plan 4: Changing Clustered Index (not Primary Key) and creating Primary Key as a Non-Clustered Index

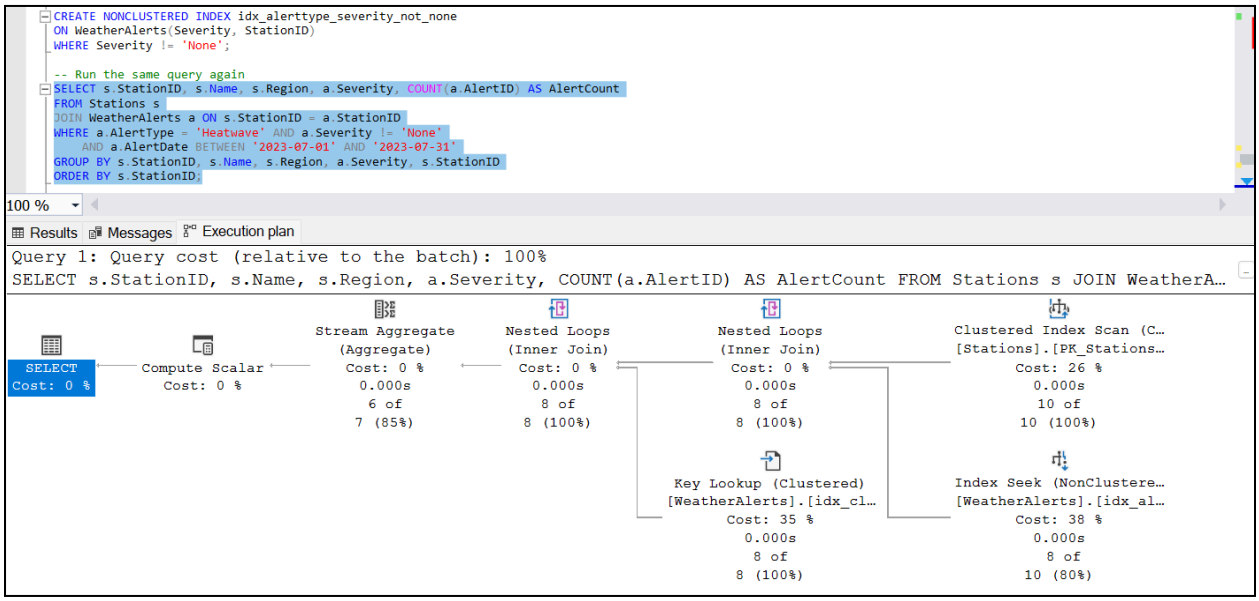
1. ***Screenshot of the query plan***



2. ***Changes made in the schema:***
   Changing Clustered Index from WeatherAlerts(AlertID) to WeatherAlerts(Severity) but Keep WeatherAlerts(AlertID) as a Primary Key
3. ***Optimizer behavior:***
   Now, there is a Primary Key Clustered Index on Stations(StationID), a Primary Key Non-Clustered Index on WeatherAlerts(AlertID), and a Clustered Index on WeatherAlerts(Severity), the query optimizer chooses to use:
   - The Clustered Index Seek on the `'WeatherAlerts'` table instead of Clustered Index Scan because the WHERE clause actually seeks for the Severity, which is the Clustered Index now, then Sort the data
   - Keep using Clustered Index Seek on the `'Stations'` table as Plan 2 and 3
   - Nested Loop (Inner Join) to join those 2 tables together because the dataset has only 8 rows, so the query optimizer chooses to use the Nested Loop instead of Merge Join, even though those 2 datasets are already sorted by StationID
   - The datasets after joining are already sorted, so the query optimizer chooses to use `'Stream Aggregate'` for the `'AlertCount'`
4. ***Cost and time:***
   - `Estimated Subtree Cost`: 0.0191306
   - `CPU Time` (stats): 0 ms
   - `Elapsed Time` (stats): 21 ms
   - `Actual Execution Time` (Elapsed Time): 95 ms

Query Plan 5: Adding the Check Constraint and creating a Composite Non-Clustered Index that excludes Severity = 'None'

1. ***Screenshot of the query plan***



2. ***Changes made in the schema:***
   Add the Check Constraint on WeatherAlerts(Severity) and Composite Non-Clustered Index on WeatherAlerts(Severity, StationID) that excludes Severity = 'None'
3. ***Optimizer behavior:***
   Now, there is a Primary Key Clustered Index on Stations(StationID), a Primary Key Non-Clustered Index on WeatherAlerts(AlertID), Clustered Index on WeatherAlerts(Severity), Check Constraint on WeatherAlerts(Severity), and Composite Non-Clustered Index on WeatherAlerts(Severity, StationID) that exclude Severity = 'None', the query optimizer chooses to use:
   - The Clustered Index Scan on the `'Stations'` table stead of Clustered Index Seek
   - The Index Seek on `'WeatherAlerts'` table because the Composite Non-Clustered Index on WeatherAlerts(Severity, StationID) that already exclude Severity = 'None'
   - Nested Loop (Inner Join) to join those 2 tables together because the dataset has only 8 rows, so the query optimizer choose to use the Nested Loop instead of Merge Join, even though those 2 datasets are already sorted by StationID
   - The Key Lookup on the `'WeatherAlerts'` before Nested Loop (Inner Join) to join those datasets again
   - The datasets after joining are already sorted, so the query optimizer choose to use `'Stream Aggregate'` for the `'AlertCount'`
4. ***Cost and time:***
   - `Estimated Subtree Cost`: 0.0124819
   - `CPU Time` (stats): 0 ms
   - `Elapsed Time` (stats): 19 ms
   - `Actual Execution Time` (Elapsed Time): 89 ms

Summary:

Query Plan	Schema Change Description	Estimated Subtree Cost	Actual Execution Time	Conclusion
1	Don't modify anything and run the base query	0.0360782	613 ms	Full Table Scan use the higher resources
2	Adding Primary Key Clustered Index in the Stations table	0.0192801	125 ms	- The Primary Key helps with the Physically ordered the data with help to improve the I/O - The index helps to filter the rows faster and reduce the cost
3	Adding Primary Key Clustered Index in the WeatherAlerts table, creating a Non-Clustered Index on the Severity	0.0192801	94 ms	More indexes on the joining tables help to filter the rows faster
4	Changing Clustered Index (not Primary Key) on the target column and creating Primary Key as a Non-Clustered Index	0.0191306	95 ms	Having the Clustered Index on the target help to change the query plan to seek rather than scan, it possible to filter the rows faster if there are more rows of data and also reduce the cost
5	Adding the Check Constraint and creating a Composite Non-Clustered Index that excludes Severity = 'None'	0.0124819	89 ms	- Best performance so far after exclude the data that we want to filter it out - The Composite Non-Clustered Index helps to filter the row faster and also reduce the cost

Additional Screenshot of Support Data:

Query Plan 1:

- **Query Optimizer decides to use:**
  - Table Scan on WeatherAlerts table

Table Scan

Scan rows from a table.

Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	65
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0033535 (9%)
Estimated Subtree Cost	0.0033535
Estimated CPU Cost	0.0002285
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Number of Rows to be Read	65
Estimated Row Size	44 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	4

  - Clustered Index Seek on Stations table

Table Scan

Scan rows from a table.

Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	10
Actual Number of Rows for All Executions	10
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.003293 (9%)
Estimated CPU Cost	0.000168
Estimated Subtree Cost	0.003293
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	10
Estimated Number of Rows Per Execution	10
Estimated Number of Rows to be Read	10
Estimated Row Size	46 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	5

- Hash Match (Inner Join)

Hash Match

Use each row from the top input to build a hash table, and each row from the bottom input to probe into the hash table, outputting all matching rows.

Physical Operation	Hash Match
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0180697 (50%)
Estimated I/O Cost	0
Estimated CPU Cost	0.017925
Estimated Subtree Cost	0.0247162
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	56 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	3

- Sort (before using Stream Aggregate)

Sort

Sort the input.

Physical Operation	Sort
Logical Operation	Sort
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0113988 (32%)
Estimated I/O Cost	0.0112613
Estimated CPU Cost	0.0001375
Estimated Subtree Cost	0.036115
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	56 B
Actual Rebinds	1
Actual Rewinds	0
Node ID	2

- Stream Aggregate

Stream Aggregate

Compute summary values for groups of rows in a suitably sorted stream.

Physical Operation	Stream Aggregate
Logical Operation	Aggregate
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	6
Actual Number of Batches	0
Estimated Operator Cost	0.0000087 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000088
Estimated Subtree Cost	0.0361237
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	7.95153
Estimated Number of Rows Per Execution	7.95153
Estimated Row Size	60 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	1

- Estimated Cost

SELECT

Cached plan size	64 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0.0360782
Memory Grant	1600 KB
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	7.95153

- Execution time (Stats)

```
SQL Server Execution Times:
  CPU time = 0 ms,  elapsed time = 97 ms.
```

- Actual Execution Time

Aggregate Status

Connection failures	
Elapsed time	00:00:00.613
Finish time	4/21/2025 8:04:17 PM
Name	LAPTOP-4HVIP9VM
Rows returned	6
Start time	4/21/2025 8:04:16 PM
State	Open

Query Plan 2:

- **Query Optimizer decides to use:**
  - Table Scan on WeatherAlerts table

Table Scan

Scan rows from a table.

Physical Operation	Table Scan
Logical Operation	Table Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	65
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0033535 (17%)
Estimated Subtree Cost	0.0033535
Estimated CPU Cost	0.0002285
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Number of Rows to be Read	65
Estimated Row Size	44 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	4

  - Sort after Table Scan on WeatherAlearts table

Sort

Sort the input.

Physical Operation	Sort
Logical Operation	Sort
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0115404 (60%)
Estimated I/O Cost	0.0112613
Estimated CPU Cost	0.0001375
Estimated Subtree Cost	0.0148939
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	23 B
Actual Rebinds	1
Actual Rewinds	0
Node ID	3

- Clustered Index Seek on Stations Table

Clustered Index Seek (Clustered)

Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	8
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0043898 (23%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0043898
Estimated CPU Cost	0.0001581
Estimated Number of Executions	8
Number of Executions	8
Estimated Number of Rows for All Executions	8
Estimated Number of Rows to be Read	1
Estimated Number of Rows Per Execution	1
Estimated Row Size	167 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	5

- Nested Loop (Inner Join)

Nested Loops

For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.

Physical Operation	Nested Loops
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0000335 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000334
Estimated Subtree Cost	0.0193172
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	177 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	2

- Stream Aggregate

Stream Aggregate

Compute summary values for groups of rows in a suitably sorted stream.

Physical Operation	Stream Aggregate
Logical Operation	Aggregate
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	6
Actual Number of Batches	0
Estimated Operator Cost	0.0000084 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000084
Estimated Subtree Cost	0.0193256
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	7.29088
Estimated Number of Rows Per Execution	7.29088
Estimated Row Size	181 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	1

- Estimated Cost

SELECT

Cached plan size	40 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0.0192801
Memory Grant	1024 KB
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	7.29088

- Execution time (Stats)

```
SQL Server Execution Times:
  CPU time = 0 ms,  elapsed time = 25 ms.
```

- Actual Execution Time

Aggregate Status

Connection failures	
Elapsed time	00:00:00.125
Finish time	4/21/2025 8:05:56 PM
Name	LAPTOP-4HVIP9VM
Rows returned	6
Start time	4/21/2025 8:05:56 PM
State	Open

Query Plan 3:

- **Query Optimizer decides to use:**
  - Clustered Index Scan on WeatherAlerts table

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	65
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.0033535 (17%)
Estimated Subtree Cost	0.0033535
Estimated CPU Cost	0.0002285
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Number of Rows to be Read	65
Estimated Row Size	44 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	4

  - Sort after Clustered Index Scan on WeatherAlearts table

Sort

Sort the input.

Physical Operation	Sort
Logical Operation	Sort
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0115404 (60%)
Estimated I/O Cost	0.0112613
Estimated CPU Cost	0.0001375
Estimated Subtree Cost	0.0148939
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	23 B
Actual Rebinds	1
Actual Rewinds	0
Node ID	3

- Clustered Index Seek on Stations table

Clustered Index Seek (Clustered)

Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	8
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0043898 (23%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0043898
Estimated CPU Cost	0.0001581
Estimated Number of Executions	8
Number of Executions	8
Estimated Number of Rows for All Executions	8
Estimated Number of Rows to be Read	1
Estimated Number of Rows Per Execution	1
Estimated Row Size	167 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	5

- Nested Loop (Inner Join)

Nested Loops

For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.

Physical Operation	Nested Loops
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0000335 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000334
Estimated Subtree Cost	0.0193172
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	177 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	2

- Stream Aggregate

Stream Aggregate

Compute summary values for groups of rows in a suitably sorted stream.

Physical Operation	Stream Aggregate
Logical Operation	Aggregate
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	6
Actual Number of Batches	0
Estimated Operator Cost	0.0000084 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000084
Estimated Subtree Cost	0.0193256
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	7.29088
Estimated Number of Rows Per Execution	7.29088
Estimated Row Size	181 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	1

- Estimated Cost

SELECT

Cached plan size	40 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0.0192801
Memory Grant	1024 KB
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	7.29088

- Execution time (Stats)

```
SQL Server Execution Times:
  CPU time = 0 ms,  elapsed time = 22 ms.
```

- Actual Execution time

Aggregate Status

Connection failures	
Elapsed time	00:00:00.094
Finish time	4/21/2025 8:06:49 PM
Name	LAPTOP-4HVIP9VM
Rows returned	6
Start time	4/21/2025 8:06:49 PM
State	Open

Query Plan 4:

- **Query Optimizer decides to use:**
  - Clustered Index Seek on WeatherAlerts table

Clustered Index Seek (Clustered)

Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	8
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0032908 (17%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0032908
Estimated CPU Cost	0.0001658
Estimated Number of Executions	1
Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows to be Read	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	44 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	4

  - Sort after Clustered Index Seek on WeatherAlearts table

Sort

Sort the input.

Physical Operation	Sort
Logical Operation	Sort
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0114082 (60%)
Estimated I/O Cost	0.0112613
Estimated CPU Cost	0.0001375
Estimated Subtree Cost	0.014699
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	23 B
Actual Rebinds	1
Actual Rewinds	0
Node ID	3

- Clustered Index Seek on Stations table

Clustered Index Seek (Clustered)

Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	8
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0043898 (23%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0043898
Estimated CPU Cost	0.0001581
Estimated Number of Executions	8
Number of Executions	8
Estimated Number of Rows for All Executions	8
Estimated Number of Rows to be Read	1
Estimated Number of Rows Per Execution	1
Estimated Row Size	167 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	5

- Nested Loop (Inner Join)

Nested Loops

For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.

Physical Operation	Nested Loops
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0000334 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000334
Estimated Subtree Cost	0.0191222
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	177 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	2

- Stream Aggregate

Stream Aggregate

Compute summary values for groups of rows in a suitably sorted stream.

Physical Operation	Stream Aggregate
Logical Operation	Aggregate
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	6
Actual Number of Batches	0
Estimated Operator Cost	0.0000084 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000084
Estimated Subtree Cost	0.0191306
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	7.29088
Estimated Number of Rows Per Execution	7.29088
Estimated Row Size	181 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	1

- Estimated Cost

SELECT

Cached plan size	48 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0.0191306
Memory Grant	1024 KB
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	7.29088

- Execution time (Stats)

```
SQL Server Execution Times:
  CPU time = 0 ms,  elapsed time = 21 ms.
```

- Actual Execution time

Aggregate Status

Connection failures	
Elapsed time	00:00:00.095
Finish time	4/21/2025 8:07:35 PM
Name	LAPTOP-4HVIP9VM
Rows returned	6
Start time	4/21/2025 8:07:35 PM
State	Open

Query Plan 5:

- **Query Optimizer decides to use:**
  - Clustered Index Seek on Stations table

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	10
Actual Number of Rows for All Executions	10
Actual Number of Batches	0
Estimated I/O Cost	0.003125
Estimated Operator Cost	0.003293 (26%)
Estimated CPU Cost	0.000168
Estimated Subtree Cost	0.003293
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	10
Estimated Number of Rows Per Execution	10
Estimated Number of Rows to be Read	10
Estimated Row Size	167 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	4

  - Index Seek on WeatherAlearts table

Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	8
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.004706 (38%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.004706
Estimated CPU Cost	0.0001581
Estimated Number of Executions	10
Number of Executions	10
Estimated Number of Rows for All Executions	10
Estimated Number of Rows to be Read	1
Estimated Number of Rows Per Execution	1
Estimated Row Size	23 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	5

- Nested Loop (Inner Join)

Nested Loops

For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.

Physical Operation	Nested Loops
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0000418 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000418
Estimated Subtree Cost	0.0080408
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	181 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	3

- Key Lookup

Key Lookup (Clustered)

Uses a supplied clustering key to lookup on a table that has a clustered index.

Physical Operation	Key Lookup
Logical Operation	Key Lookup
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows Read	8
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0043898 (35%)
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001581
Estimated Subtree Cost	0.0043898
Number of Executions	8
Estimated Number of Executions	8
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	1
Estimated Row Size	64 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	7

- Nested Loop

Nested Loops

For each row in the top (outer) input, scan the bottom (inner) input, and output matching rows.

Physical Operation	Nested Loops
Logical Operation	Inner Join
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	8
Actual Number of Batches	0
Estimated Operator Cost	0.0000334 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000334
Estimated Subtree Cost	0.012464
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	8
Estimated Number of Rows Per Execution	8
Estimated Row Size	198 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	2

- Stream Aggregate

Stream Aggregate

Compute summary values for groups of rows in a suitably sorted stream.

Physical Operation	Stream Aggregate
Logical Operation	Aggregate
Actual Execution Mode	Row
Estimated Execution Mode	Row
Actual Number of Rows for All Executions	6
Actual Number of Batches	0
Estimated Operator Cost	0.0000179 (0%)
Estimated I/O Cost	0
Estimated CPU Cost	0.0000084
Estimated Subtree Cost	0.0124819
Number of Executions	1
Estimated Number of Executions	1
Estimated Number of Rows for All Executions	7.29088
Estimated Number of Rows Per Execution	7.29088
Estimated Row Size	181 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	1

- Estimated Cost

SELECT	
Cached plan size	48 KB
Estimated Operator Cost	0 (0%)
Degree of Parallelism	1
Estimated Subtree Cost	0.0124819
Estimated Number of Rows for All Executions	0
Estimated Number of Rows Per Execution	7.29088

- Execution time (Stats)

```
SQL Server Execution Times:
  CPU time = 0 ms,  elapsed time = 19 ms.
```

- Actual Execution Time