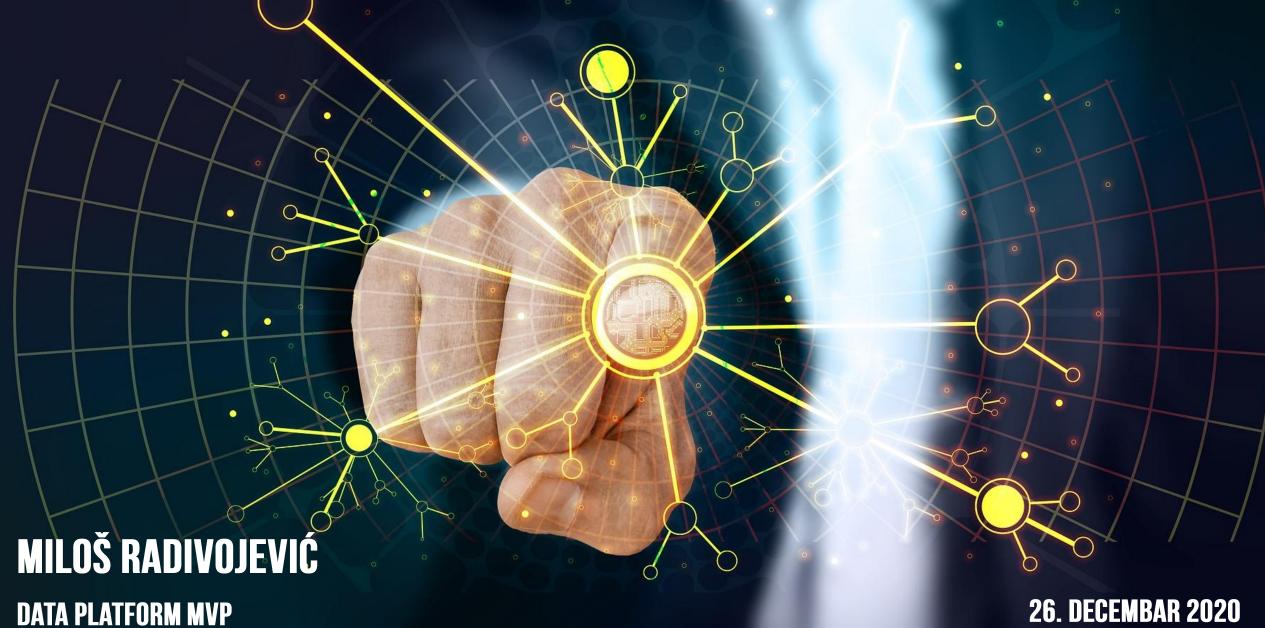
INTELIGENTNO PROCESIRANJE U SQL SERVERU 2019

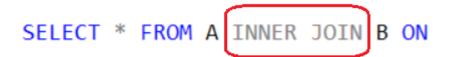


ADAPTIVE JOIN

- Enterprise Edition Feature
- Improvement that enables the optimizer to delay the choice of the join method (hash join or nested loops) at runtime
- A new operator Adaptive Join

JOIN IMPLEMENTATION

Logical Operation



Physical Operators







NESTED LOOPS JOIN

```
-- Nested Loops Join
    SELECT o.SalesOrderID, c.LastName
     FROM Person Person c
     INNER JOIN Sales.SalesOrderHeader o ON o.CustomerID = c.BusinessEntityID
     WHERE c.BusinessEntityID = 14501;
112 % +
Results Messages Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT o.SalesOrderID, c.LastName FROM Person.Person c INNER JOIN Sales.SalesOrderHeader o ON o.C.
               Nested Loops
                                   Clustered Index Seek (Clustered)
                                  [Person] . [PK Person BusinessEntityI ...
                (Inner Join)
  SELECT
                                             Cost: 50 %
 Cost: 0 %
           Actual Number of Rows for All Executions
           Estimated Number of Rows Per Execution
                                               1,65926
           Estimated Row Size
                                                 65 B
           Estimated Data Size
                                                108 B
                                      Index Seek (NonClustered)
                                  [SalesOrderHeader].[IX SalesOrderHe...
                                             Cost: 50 %
                                               0.000s
                                                2 of
                                              2 (100%)
```

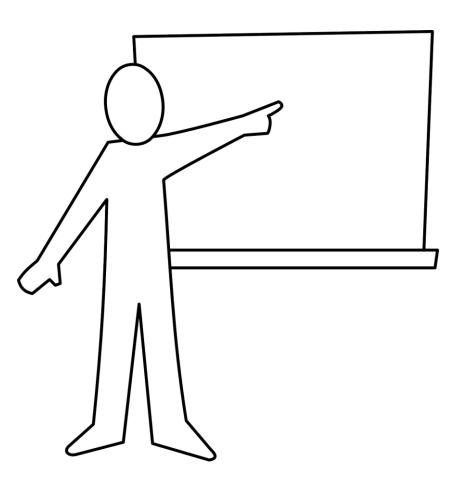
MERGE JOIN

```
-Merge Join
    SELECT o.SalesOrderID, o.OrderDate, o.CurrencyRateID, c.*
     FROM Person Person c
     INNER JOIN Sales.SalesOrderHeader o ON o.CustomerID = c.BusinessEntityID
112 %
Results Messages Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT o.SalesOrderID, o.OrderDate, o.CurrencyRateID, c.* FROM Person.Person c INNER JOIN Sales.Sale
                  Merge Join
                                       Clustered Index Scan (Clustered)
                                     [Person].[PK Person BusinessEntityI...
                  Inner Join)
  SELECT
                                                Cost: 50 %
                  Cost: 2 %
                   0.028s
 Cost: 0 %
                                                  0.008s
           Actual Number of Rows for All Executions
                                                 17463
           Estimated Number of Rows Per Execution
                                                17462.7
           Estimated Row Size
                                                8293 B
           Estimated Data Size
                                                138 MB
                                                                    Clustered Index Scan (Clustered)
                                                   Sort
                                                                  [SalesOrderHeader].[PK SalesOrderHe...
                                                Cost: 38 %
                                                                              Cost: 10 %
                                                  0.017s
                                                                               0.003s
                                                 17464 of
                                                                               31465 of
                                                31465 (55%)
                                                                             31465 (100%)
```

HASH JOIN

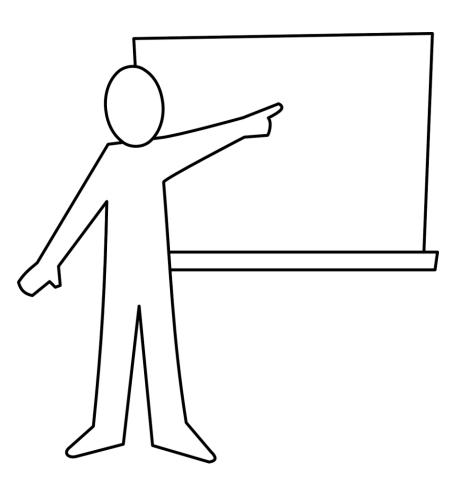
```
--Hash Join
   □SELECT o.SalesOrderID, o.OrderDate, c.LastName
     FROM Person Person c
     INNER JOIN Sales.SalesOrderHeader o ON o.CustomerID = c.BusinessEntityID
112 % -
Results Messages Execution plan
Query 1: Query cost (relative to the batch): 100%
SELECT o.SalesOrderID, o.OrderDate, c.LastName FROM Person.Person c INNER JOIN Sales.SalesOrderHeade
                    猖
                                         Index Scan (NonClustered)
                 Hash Match
                                     [Person].[IX Person LastName FirstN...
                 (Inner Join) @
  SELECT
                 Cost: 57 %
                                                Cost: 7 %
                                                 0 0015
                   0.0124
Cost: 0 %
           Actual Number of Rows for All Executions
                                                17463
                                               17462.7
           Estimated Number of Rows Per Execution
           Estimated Row Size
                                                 73 B
           Estimated Data Size
                                               1245 KB
                                      Clustered Index Scan (Clustered)
                                     [SalesOrderHeader].[PK SalesOrderHe...
                                                Cost: 36 %
                                                  0.003s
                                                 31465 of
                                               31465 (100%)
```

JOIN AND PARAMETER SENSITIVE QUERIES



DEMO

- 4. Join operator adaptive join
- Has both hash join and nested loops join selection at runtime
- Starts as a hash join and scans the input
 - Estimated Number of Rows <Adaptive Threshold => changes to Nested Loop
 Join
 - Estimated Number of Rows> = Adaptive Threshold => continues as a hash join
- AJ handles parameter sensitive queries better, but cannot solve all problems where the wrong join operator is selected
- It only works in batch mode



DEMO

OSTRESS

RML Utilities (ostress tool)

https://www.microsoft.com/en-us/download/details.aspx?id=4511

RML Utilities for SQL Server (x64) CU4

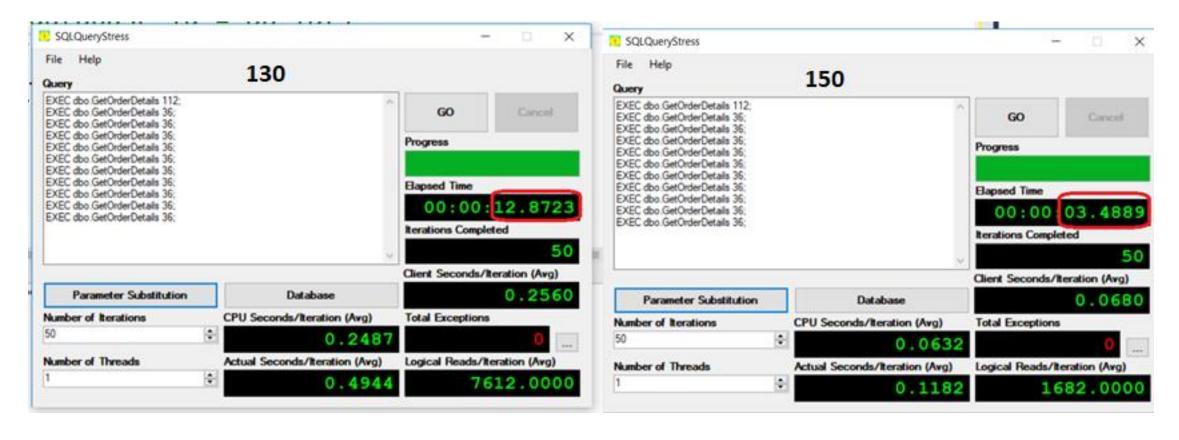
Important! Selecting a language below will dynamically change the complete page content to that language.

Language:

English

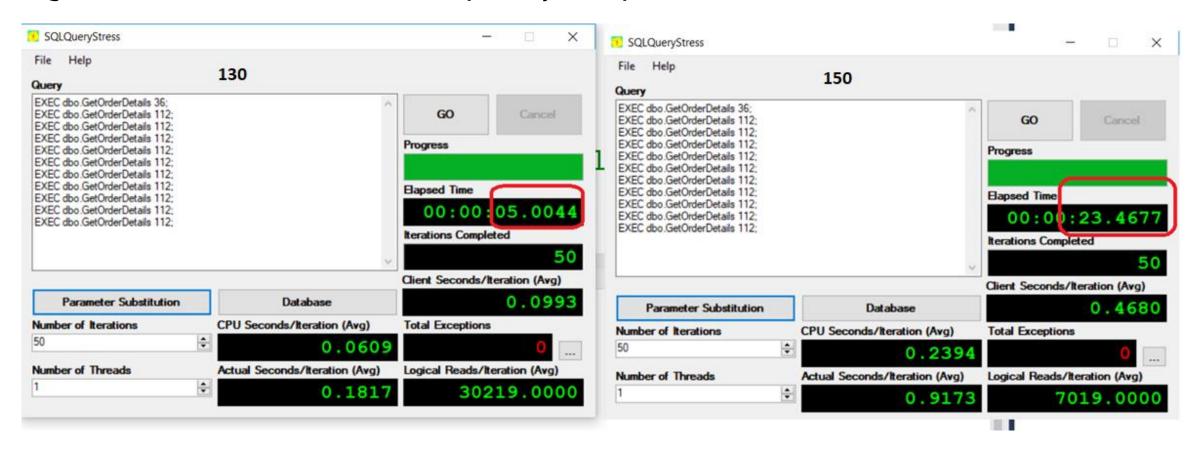
Download

Q: Is it better with the new adaptive join operator?



A: YES!!! The query is 4x faster under the CL 150!

Q: Is it better with the new adaptive join operator?



A: Actually NO - the query runs 5x slower under the CL 150!

AJ SETTINGS IN SQL SERVER 2017

Default: OFF

Enable :

```
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_BATCH_MODE_ADAPTIVE_JOINS = OFF;
```

- Disable :
 - At the database level:

```
ALTER DATABASE SCOPED CONFIGURATION SET DISABLE_BATCH_MODE_ADAPTIVE_JOINS = ON;
```

• At the statement level:

```
OPTION(USE HINT('DISABLE_BATCH_MODE_ADAPTIVE_JOINS'));
```

AJ SETTINGS IN SQL SERVER 2019

Default: OFF

Enable:

ALTER DATABASE SCOPED CONFIGURATION SET BATCH_MODE_ADAPTIVE_JOINS = ON;

- Disable:
 - At the database level:

ALTER DATABASE SCOPED CONFIGURATION SET BATCH_MODE_ADAPTIVE_JOINS = OFF;

At the statement level

```
OPTION(USE HINT('DISABLE_BATCH_MODE_ADAPTIVE_JOINS'));
```

CONCLUSION

- 4. Join operator, can improve the performance of parameter-sensitive queries
- It can also cause regression
- Therefore test, test, test!
- Enterprise Edition feature
- It only works in batch mode