# An Introduction to SQL Server for Data Scientists

**Gavin Payne**
gavin@gavinpayne.co.uk
@GavinPayneUK

# Gavin Payne

- 20 years' experience working with SQL Server
- Microsoft Certified Architect for SQL Server
- Microsoft Certified Master for SQL Server

# Agenda

- What is SQL Server?
- Processing a query
- How to make queries run faster
- Its role as a machine learning platform
- Questions

# What is SQL Server?

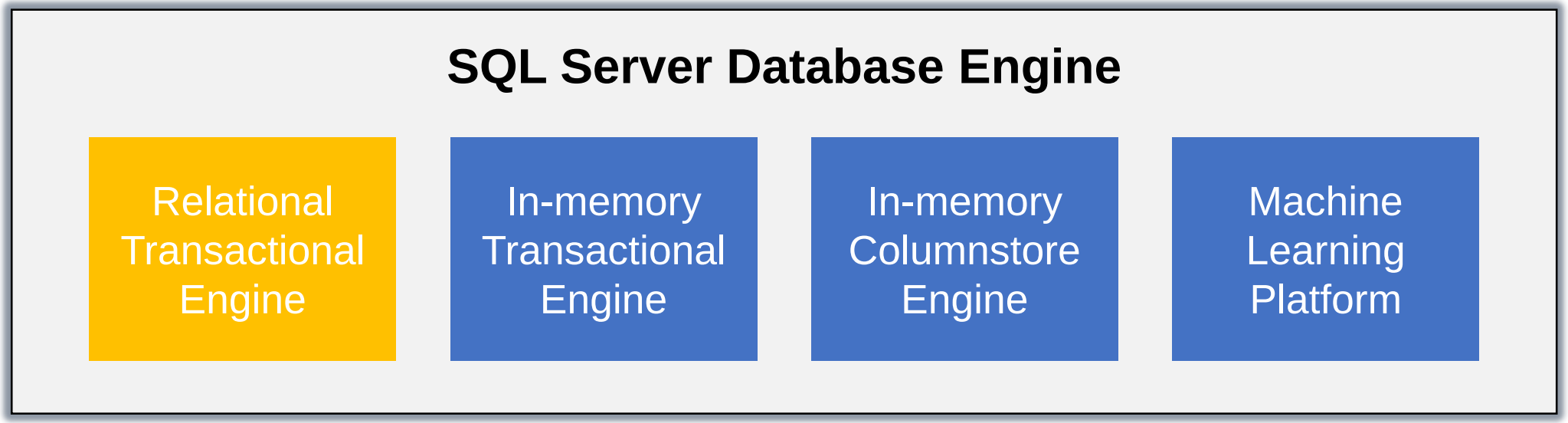
# What is SQL Server?

- **Microsoft's relational data platform**
- V1.0 released in 1989, major rewrite in 2005, latest released in 2017


- **Database engine ("SQL Server")**
- SQL Server Integration Services
- SQL Server Analysis Services
- SQL Server Reporting Services
- Master Data Management
- Data Quality Services

# The SQL Server Database Engine

- **Relational transactional database engine** (ACID compliant)
- Used by commercial and bespoke applications – and analytics
- Foundation for Azure SQL Database and Azure SQL Data Warehouse



# Using SQL Server

- **Free editions and tools**
  - SQL Server Developer Edition
  - SQL Server Express Edition
  - SQL Server Management Studio
  - SQL Operations Studio
  - Visual Studio Code mssql extension

Start Here

Start Here

Start Here



- **Commercial editions**
  - SQL Server Standard Edition
  - SQL Server Enterprise Edition

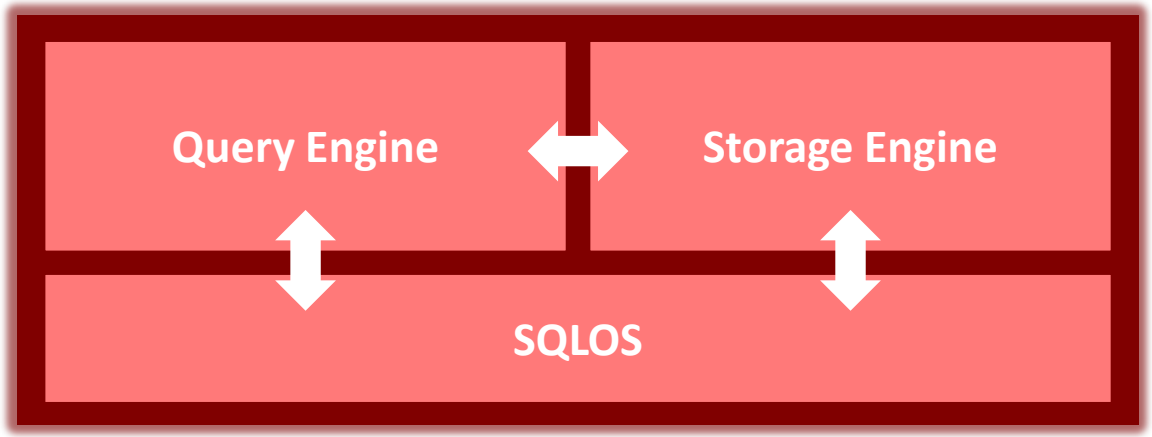
**Client and server support for:**

- Linux, MacOS, and Containers
- (and Windows)

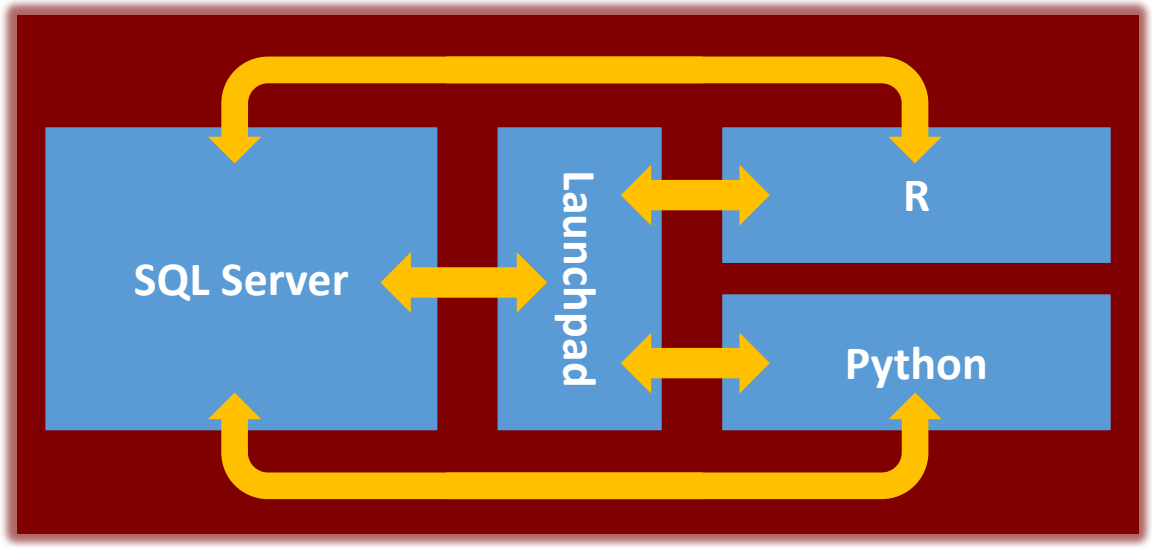
# Processing a Query

# Processing a Query

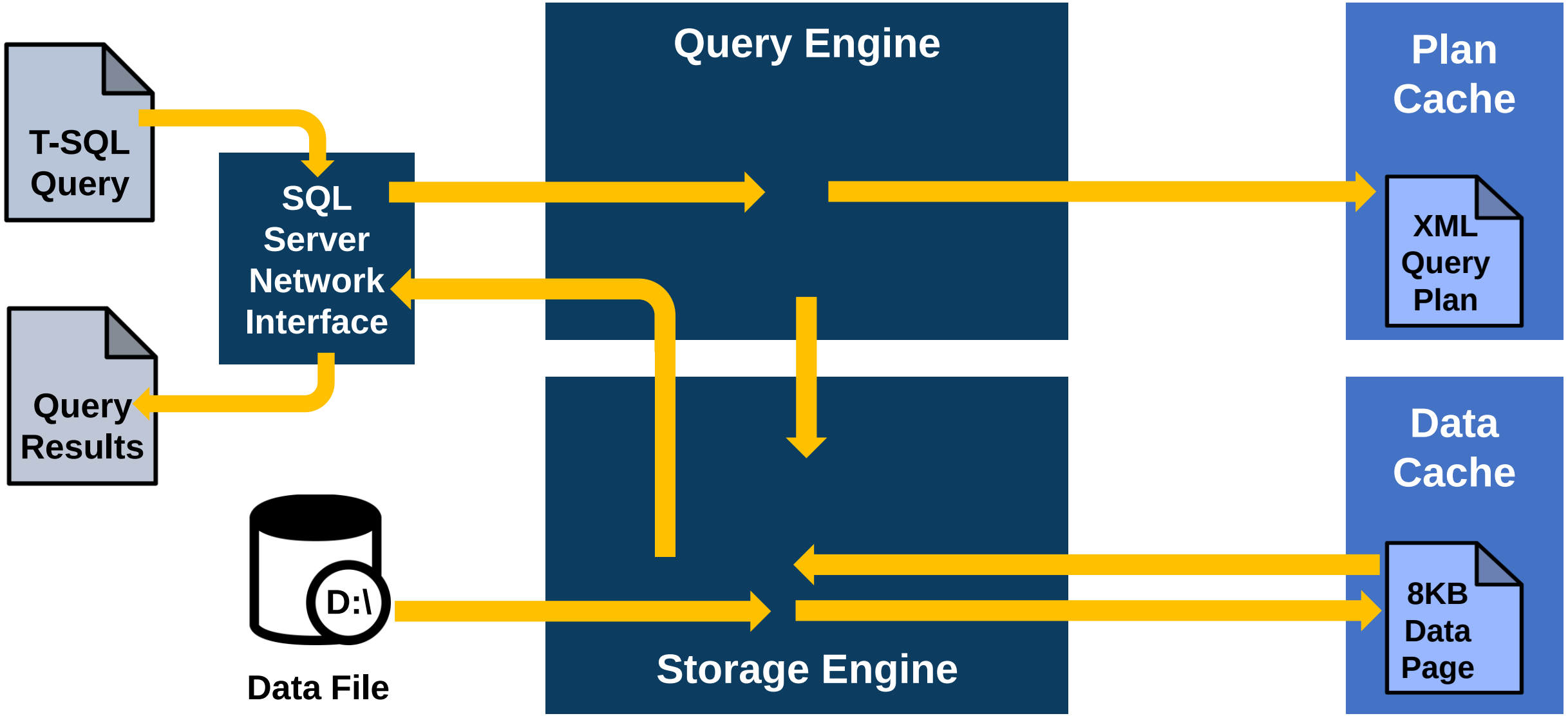
- **SQL Server Database Engine**
  - Query engine
  - Storage engine
  - SQLOS




- **Machine learning interfaces**
  - R
  - Python



# Processing a Query



# Processing a Query - Memory

- **Queries always read data from memory, never from disk**
  - Disk reads much more expensive than memory reads
- **The most common performance bottleneck**
  - 10GB of data and 40GB of memory?
  - 40GB of data and 10GB of memory?
- **Why doesn’t sqlserver.exe release memory?**
- **Limit the amount of memory used with the Max Server Memory setting**

# Processing a Query - Compute

- **Query engine determines number of processors to use**
  - Serial plan – One processor
  - Parallel plan – Multiple processors

- **Limit the number of processors per query using the MaxDOP setting**

- **By default, an expensive plan can use every processor**
  - Heavyweight analytics queries
  - Poorly indexed queries

# How to make queries run faster

# Database Indexing

- **What is an index?**

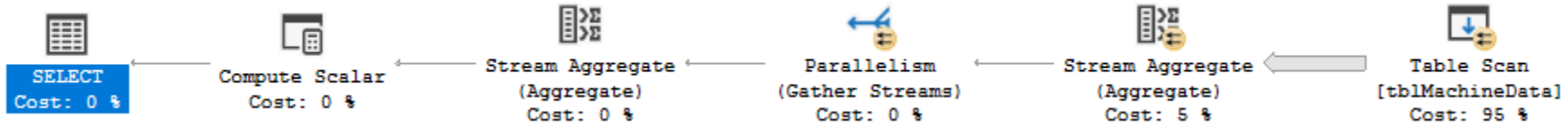
  A user-defined collection of key values based on known queries
  Known as non-clustered indexes in SQL Server

- **Indexes reduce the amount of data a query processes**
  - Fewer CPU cycles
  - Fewer disk reads

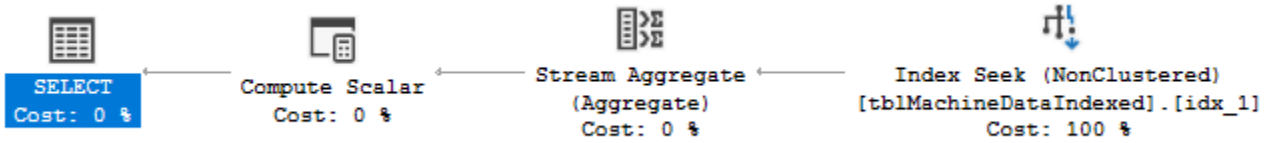
**Easiest way to tune query performance**

# Database Indexing

- **Scan** - Reads every row in the table




- **Seek** - Lookup which rows to read, then just read those



# Database Indexing

- **Is the answer always to add indexes?**

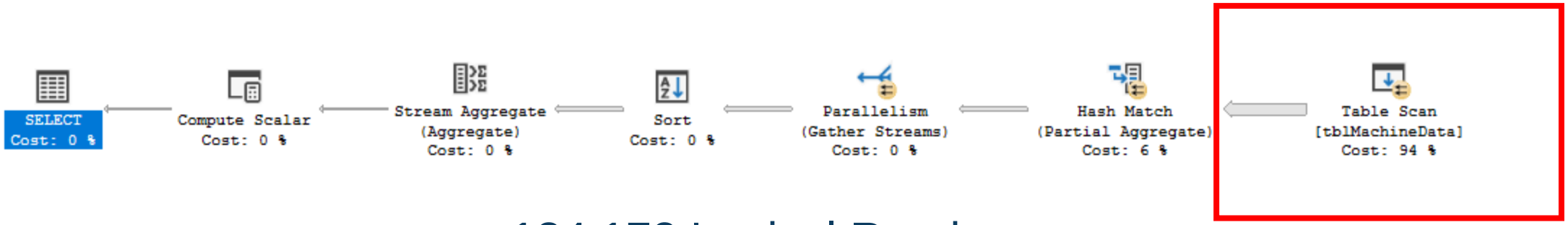
**tblMachineData**

| MachineNo | MachineLocation | ValueTime | SensorID | SensorValue |
|---|---|---|---|---|
| Bigint | Varchar(20) | DateTime | Int | Decimal |

```
select      SensorID as 'Sensor',
            min(SensorValue) as 'Min',
            avg(SensorValue) as 'Avg',
            max(SensorValue) as 'Max'
from        tblMachineData
where       MachineLocation = 'Factory 28'
group by    SensorID
```

# Database Indexing

No Indexes

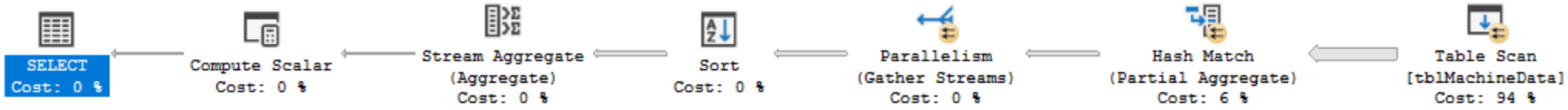



134,172 Logical Reads

2.54s Duration

# Database Indexing

```
create index idx_1 on tblMachineDataIndexed(MachineLocation)
```

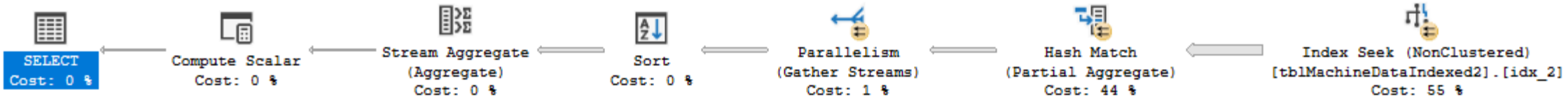



133,791 Logical Reads

2.51s Duration

# Database Indexing

```
create index idx_2 on tblMachineDataIndexed2(MachineLocation)
include (sensorID, sensorValue)
```

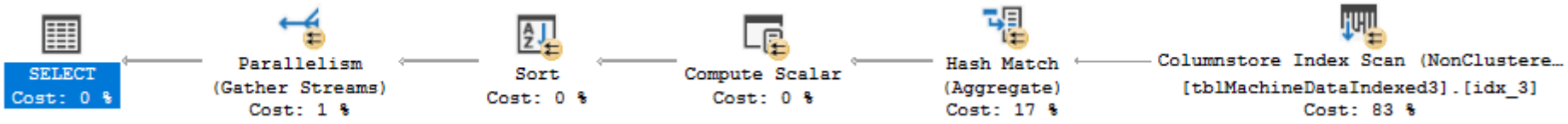



3,788 Logical Reads

**0.24s Duration**

# Database Indexing

```
create columnstore index idx_3
on tblMachineDataIndexed3(MachineLocation, sensorID, sensorValue)
```




18,715 Logical Reads

**0.02s Duration**

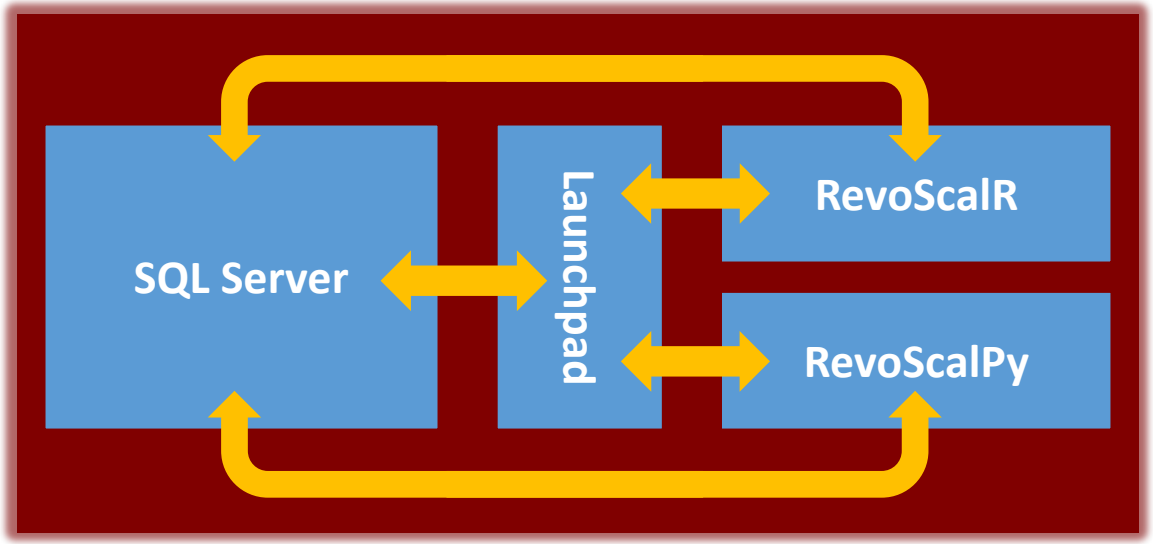
# Columnstore Indexes

- **SQL Server's internal analytics engine**
  - In-memory columnar storage and query engine
  - Replaces traditional transactional row-based engine
  - Uses existing T-SQL queries
- **What's the catch?**

# SQL Server as a Machine Learning Platform

# SQL Server Machine Learning Services

- **Taking the data science to the (SQL Server) data**
  - Processes external to SQL Server
  - Data stays within its security boundary



# Executing Machine Learning Queries

- **sp_execute_external_script**
  - Generates external calls to the R or Python runtimes
  - Complete data science language support

```
EXECUTE           sp_execute_external_script
                  @language = N'R',
                  @script =   N'model = rxUnserializeModel(lin_model);
                              automobiles_prediction = rxPredict(model, automobiles_test)
                              automobiles_pred_results <- cbind(automobiles_test, automobiles_prediction)',
                     @input_data_1 = N'
                                     SELECT       wheel_base,
                                                  length,
                                                  price
                                       FROM       Automobile_Test',
                     @input_data_1_name = N'automobiles_test',
                     @output_data_1_name = N'automobiles_pred_results',
                     @params = N'@lin_model varbinary(max)',
                     @lin_model = @lin_model_raw
      WITH RESULT SETS (("wheel_base" FLOAT, "length" FLOAT, "width" FLOAT, "height" FLOAT,
                         "curb_weight" FLOAT, "engine_size" FLOAT, "horsepower" FLOAT, "price" FLOAT, "predicted_price" FLOAT))
```

# Realtime Scoring

- **sp_rxPredict**
  - Native T-SQL scoring
  - Requires pretrained models
  - Designed to be very fast

```
DECLARE @model =     SELECT @model FROM model_table
                     WHERE model_name = 'rxLogit trained';

EXEC sp_rxPredict @model = @model, @inputData = N'SELECT * FROM data';
```

# Native Scoring

- **PREDICT**
  - *Even faster* than realtime scoring
  - Does not use any R libraries
  - Available in all installations of SQL Server

```
SELECT d.*, p.*
FROM PREDICT(MODEL = @model, DATA = dbo.iris_rx_data as d)
```

# Summary

- SQL Server is an established but modern relational data platform
- Query data always comes from memory, which comes from disk
- Tuning large analytics queries is (very) different to transactional queries
- Microsoft's R and Python services come built-in
  But so do realtime and native scoring

# Just like Jimi Hendrix …

We love to get feedback

Please complete the session feedback forms

# SQLBits - It's all about the community...

Please visit Community Corner, we are trying this year to get more people to learn about the SQL Community, equally if you would be happy to visit the community corner we'd really appreciate it.