



How the size
of character datatypes
influences
memory grants



DB

And how the engine tries to save you



About me

 Bart Vernaillen

 bart.vernaillen@d-bart.com

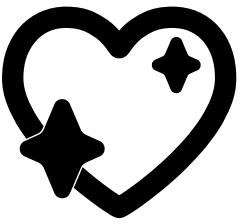
20 + years experience with:

- SQL Server
- PowerShell enthusiast



DB

Thank you, partners



PLAINSIGHT



LACO/



MONIN
Database Managed Services



u2u

tillit data
shapers

randstad
digital

inetum.^{realdolmen}
Positive digital flow



lytix

KOHERA

MICROPOLE
BELUX

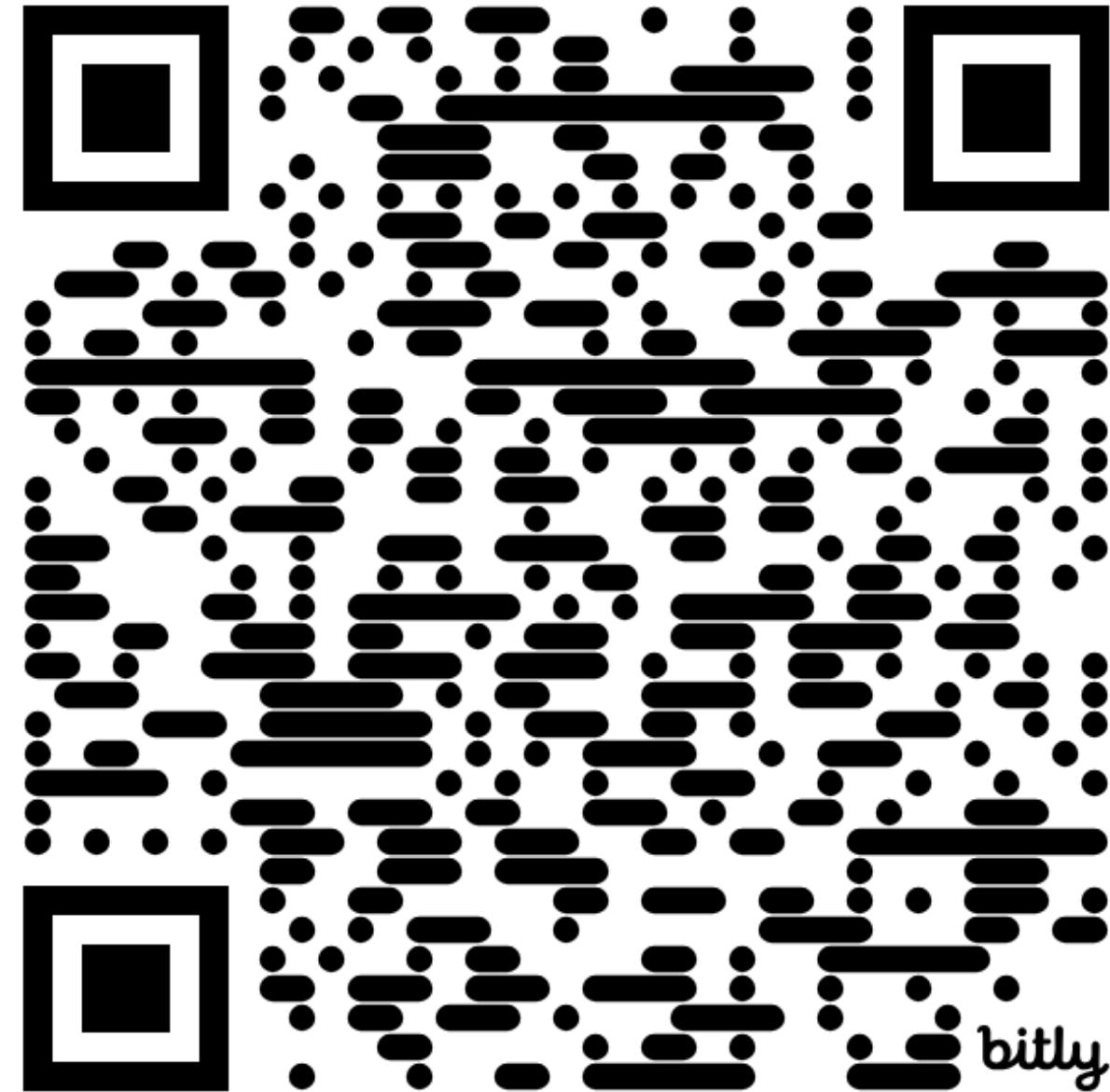
delaware

EpicData.

bmatix
Act informed

TriFinance
BEYOND ADVISORY

Session Feedback ❤️



https://bit.ly/dMC2024_SessionFeedback

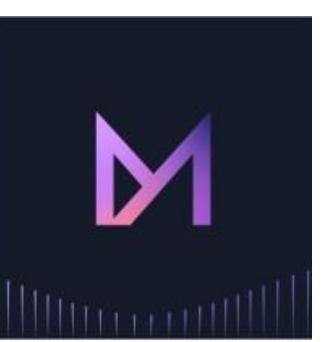


Slides:

- <https://bit.ly/3xRgiVv>
- <https://github.com/dbartv/BVernaillen-DemoDatype>



DB



Lessons learned

- Memory grants used for?
- Relation datatype & (initial) Memory Grant?
- Impact on the SQL instance?
- Impact Memory Grant Feedback?
- Remediations?



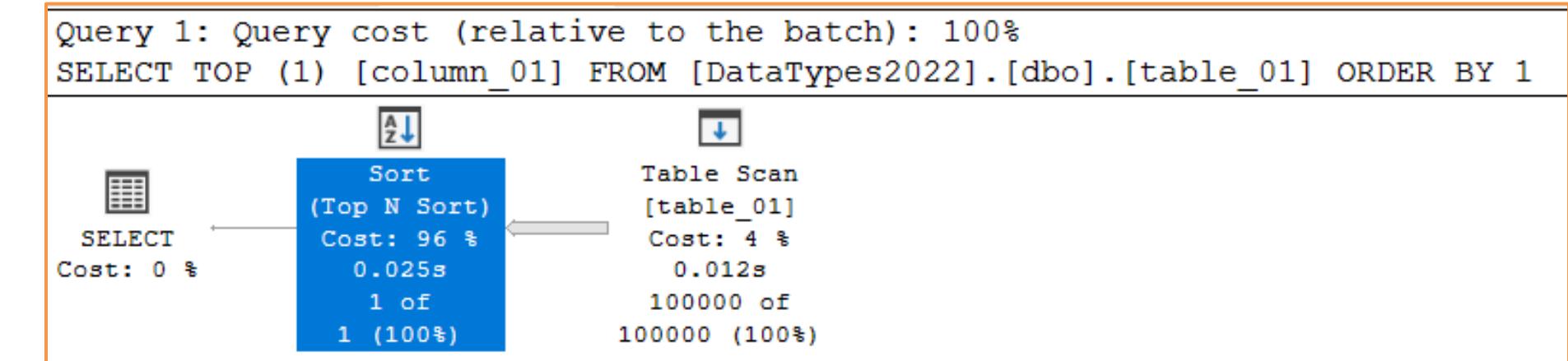
DB



Memory grants

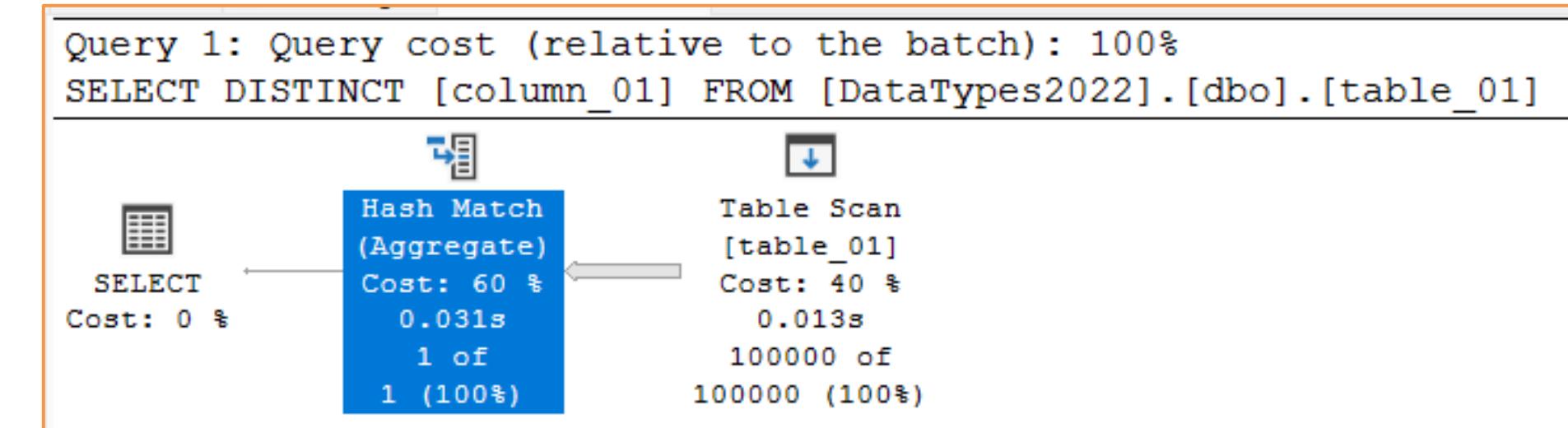
- Take place during query execution for the entire length of the execution

- Sort operator (e.g. ORDER BY)



- Hash operation

- Join large input tables (missing index?)
- Aggregates (SUM/AVG/MIN/...)
- DISTINCT
- UNION



DB



During execution

- sys.dm_exec_query_memory_grants

```
SELECT
    [session_id]
    ,[requested_memory_kb]
    ,[granted_memory_kb]
    ,[required_memory_kb]
    ,[used_memory_kb]
    ,[dop]
FROM
    [sys].[dm_exec_query_memory_grants]
ORDER BY
    grant_time desc
```

00 % ▾

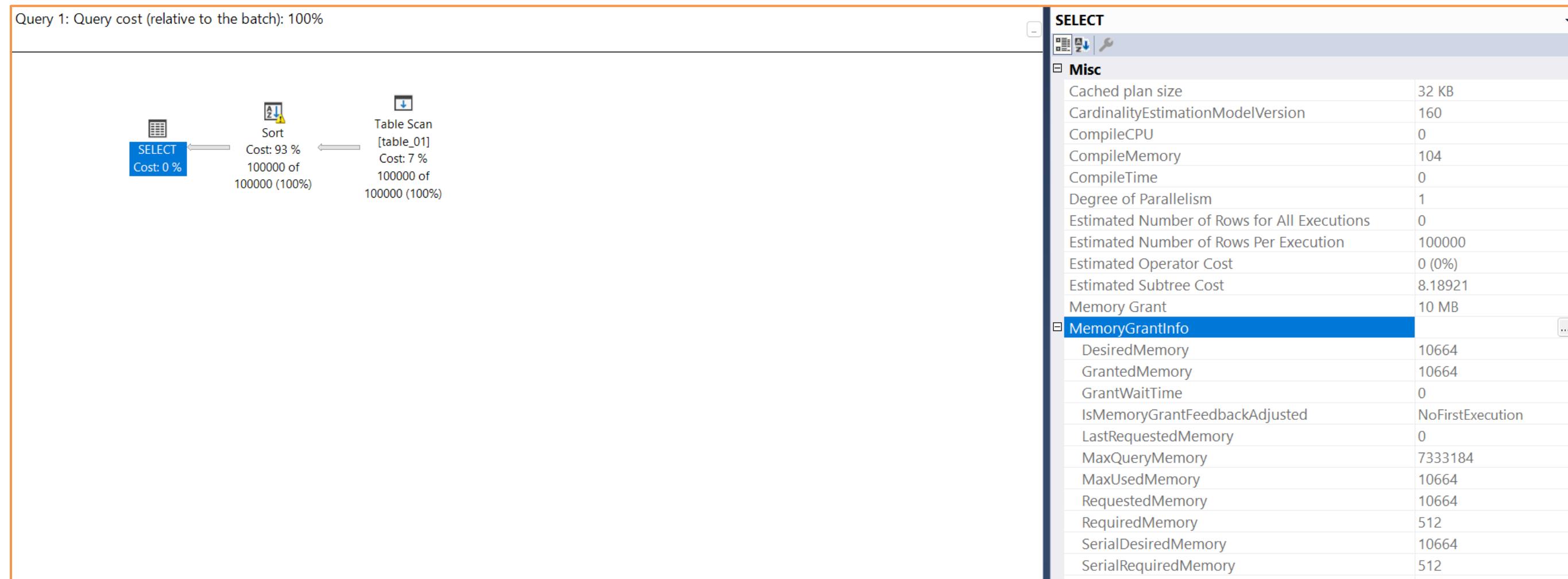
| | session_id | requested_memory_kb | granted_memory_kb | required_memory_kb | used_memory_kb | dop |
|---|------------|---------------------|-------------------|--------------------|----------------|-----|
| 1 | 74 | 1024 | 1024 | 512 | 0 | 1 |

DB



After execution

- Actual execution plan



DB



After execution

- Actual execution plan

| MemoryGrantInfo | |
|-------------------------------|------------------|
| DesiredMemory | 10664 |
| GrantedMemory | 10664 |
| GrantWaitTime | 0 |
| IsMemoryGrantFeedbackAdjusted | NoFirstExecution |
| LastRequestedMemory | 0 |
| MaxQueryMemory | 7333184 |
| MaxUsedMemory | 10664 |
| RequestedMemory | 10664 |
| RequiredMemory | 512 |
| SerialDesiredMemory | 10664 |
| SerialRequiredMemory | 512 |

```
<MemoryGrantInfo  
    SerialRequiredMemory="512"  
    SerialDesiredMemory="10664"  
    RequiredMemory="512"  
    DesiredMemory="10664"  
    RequestedMemory="10664"  
    GrantWaitTime="0"  
    MaxQueryMemory="7333184"  
    GrantedMemory="10664"  
    MaxUsedMemory="10664"  
    LastRequestedMemory="0"  
    IsMemoryGrantFeedbackAdjusted="No: First Execution"  
/>
```

DB



Demo databases design

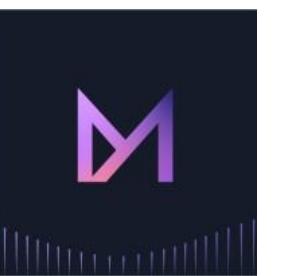
| | |
|-----|------------------|
| + 🗁 | DataTypes2017 |
| + 🗁 | DataTypes2019 |
| + 🗁 | DataTypes2022 |
| + 🗁 | DataTypes2022QSD |
| + 🗁 | DataTypesMix |

| Schema | Name | Row Count |
|--------|----------|-----------|
| dbo | table_01 | 100,000 |
| dbo | table_02 | 100,000 |
| dbo | table_03 | 100,000 |
| dbo | table_04 | 100,000 |
| dbo | table_05 | 100,000 |

| | column_01 | column_02 | column_03 | column_04 | column_05 | column_06 | column_07 | column_08 | column_09 | column_10 | column_11 | column_12 |
|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | A | B | C | D | E | F | G | H | I | J | K | L |
| 2 | A | B | C | D | E | F | G | H | I | J | K | L |
| 3 | A | B | C | D | E | F | G | H | I | J | K | L |
| 4 | A | B | C | D | E | F | G | H | I | J | K | L |
| 5 | A | B | C | D | E | F | G | H | I | J | K | L |
| 6 | A | B | C | D | E | F | G | H | I | J | K | L |
| 7 | A | B | C | D | E | F | G | H | I | J | K | L |
| 8 | A | B | C | D | E | F | G | H | I | J | K | L |
| 9 | A | B | C | D | E | F | G | H | I | J | K | L |
| 10 | A | B | C | D | E | F | G | H | I | J | K | L |

| | table_name | column_name | data_type | max_length |
|---|------------|-------------|-----------|------------|
| 1 | table_01 | column_01 | varchar | 1 |
| 2 | table_02 | column_01 | varchar | 8 |
| 3 | table_03 | column_01 | varchar | 80 |
| 4 | table_04 | column_01 | varchar | 800 |
| 5 | table_05 | column_01 | varchar | 8000 |

DB



Demo databases design

- + DataTypes2017
- + DataTypes2019
- + DataTypes2022
- + DataTypes2022QSD
- + DataTypesMix

| Schema | Name | Row Count |
|--------|----------|-----------|
| dbo | table_01 | 100,000 |
| dbo | table_02 | 100,000 |
| dbo | table_03 | 100,000 |
| dbo | table_04 | 100,000 |
| dbo | table_05 | 100,000 |

| TABLE_NAME | DATA_TYPE | max_length |
|------------|-----------|------------|
| table_01 | varchar | 1 |
| table_02 | varchar | 8 |
| table_03 | varchar | 80 |
| table_04 | varchar | 800 |
| table_05 | varchar | 8000 |

| | column_01 | column_02 | column_03 | column_04 | column_05 | column_06 | column_07 | column_08 | column_09 | column_10 | column_11 | column_12 |
|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | A | B | C | D | E | F | G | H | I | J | K | L |
| 2 | A | B | C | D | E | F | G | H | I | J | K | L |
| 3 | A | B | C | D | E | F | G | H | I | J | K | L |
| 4 | A | B | C | D | E | F | G | H | I | J | K | L |
| 5 | A | B | C | D | E | F | G | H | I | J | K | L |
| 6 | A | B | C | D | E | F | G | H | I | J | K | L |
| 7 | A | B | C | D | E | F | G | H | I | J | K | L |
| 8 | A | B | C | D | E | F | G | H | I | J | K | L |
| 9 | A | B | C | D | E | F | G | H | I | J | K | L |
| 10 | A | B | C | D | E | F | G | H | I | J | K | L |

DB



Demo

- Is written in PowerShell to (try) to prevent demo failure(s) 🙌
- Start XE session to capture ‘actual plans’
- Run queries
- Stop XE session
- Load plans in database ‘PlanUsageInfo’
- Copy query to clipboard

DB



Oversized columns

- Mismatch between defined data type and actual data
- The truck is taking parking space from other people
- The truck is under utilized



DB



Choose an appropriate size:

- Try to match character column length to the actual content



DB



Memory grant feedback

Enterprise edition only!!!

- Since SQL 2017 (batch mode only)
 - Tries to remember previous executions
 - adjusts memory grants based on the previous execution.
- SQL 2019 also for row mode
- New feature in SQL 2022 => ONLY WHEN QUERY STORE IS ENABLED!!!!
 - Feedback is persisted in the query store
 - Feedback is calculated on multiple previous executions
 - Feedback survives:
 - Plan cache eviction
 - Server reboot

DB



No worries then?



A good database design



Memory grant feedback

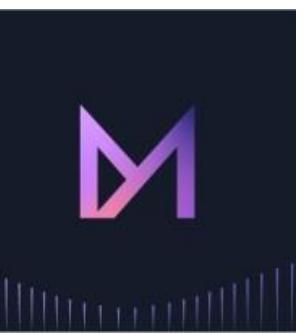
DB



Bugs found:

- Feedback is not implemented immediately
- No indication in the query plan that the MG is adjusted through Query Store
- In this demo MGF fails when queries are executed in parallel

DB



Lessons learned:

- Memory grants are used for
 - Sort operations
 - Hash operations
- The optimizer assumes that on average a column is filled for 50%
- As the length of the data type grows, the (initial) memory grant grows.
- Memory Grant Feedback can reduce the impact



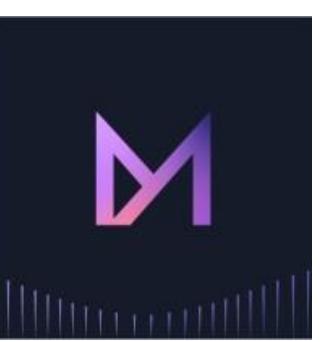
DB



What can we do?:

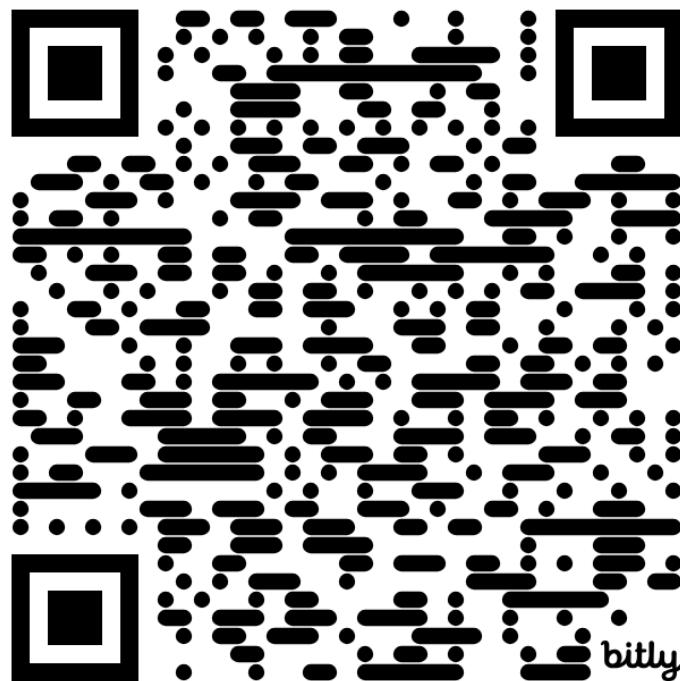
- Choose an appropriate datatype
- Choose an appropriate column length
- Avoid 'ORDER BY' if possible
- Proper indexing

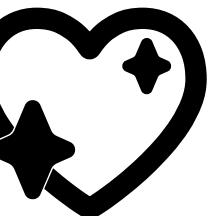
DB



Slides:

- <https://bit.ly/3xRgiVv>
- <https://github.com/dbartv/BVernaillen-DemoDatype>



Session Feedback 

DB



DB



References

- <https://sqlserverfast.com/epr/hash-match/>
- <https://sqlserverfast.com/epr/sort/>
- <https://techcommunity.microsoft.com/t5/azure-sql-blog/announcing-degree-of-parallelism-feedback-limited-preview/ba-p/3806924>
- <https://techcommunity.microsoft.com/t5/sql-server-blog/understanding-sql-server-memory-grant/ba-p/383595>
- <https://learn.microsoft.com/en-us/troubleshoot/sql/database-engine/performance/troubleshoot-memory-grant-issues>
- <https://learn.microsoft.com/en-us/sql/relational-databases/memory-management-architecture-guide?view=sql-server-ver16#memory-grant-considerations>

DB