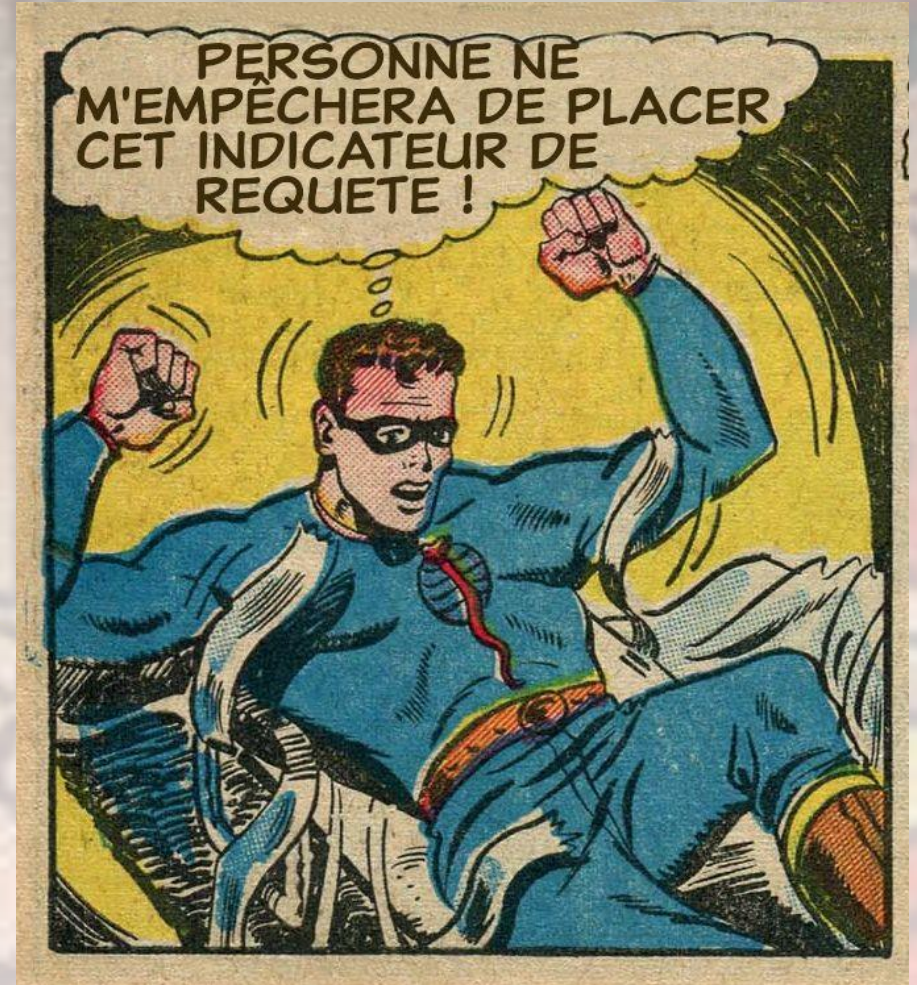


# SQL Server Query Hints (indicateurs de requêtes)

Arian PAILLON

#datafogs



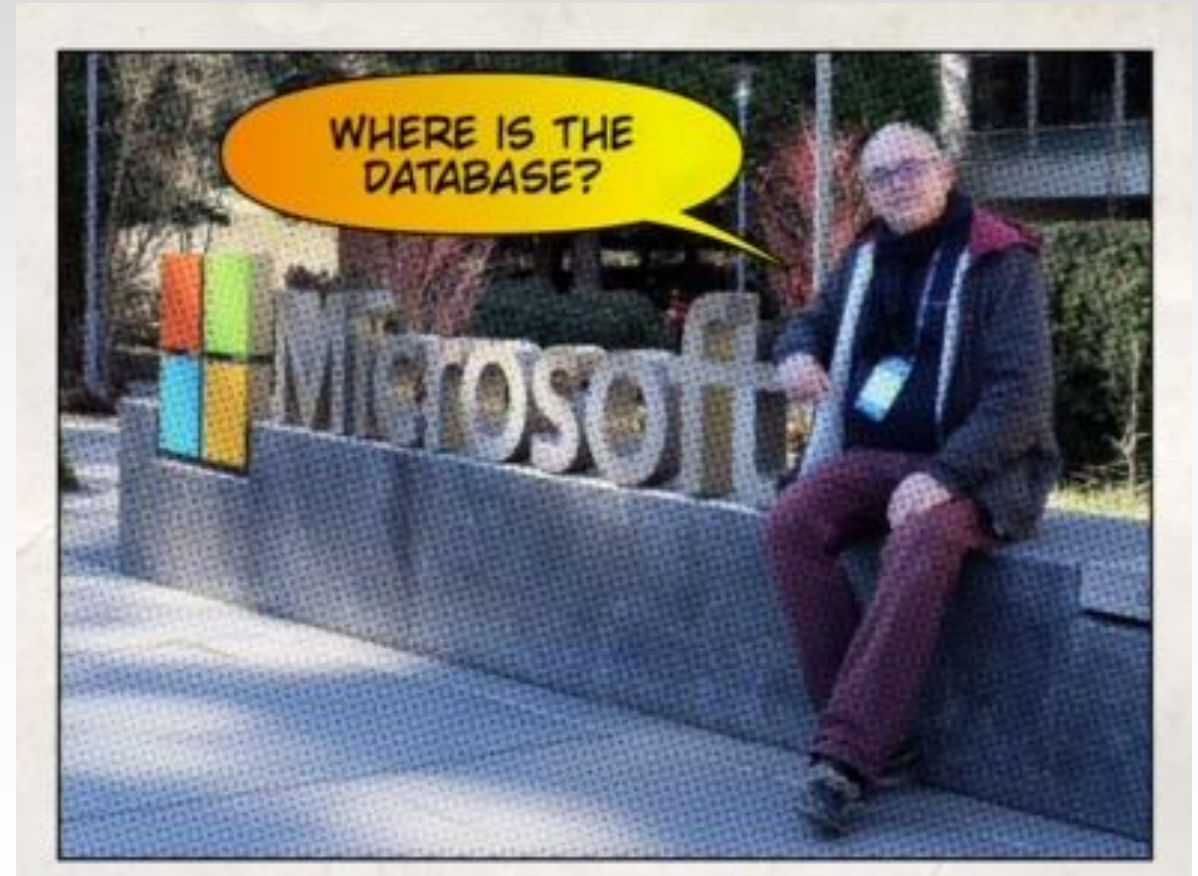


# Qui suis-je ?



## ◎ Arian PAPILLON

- > Datafly SARL (depuis 2008)
- > Expert spécialiste SQL Server avec quelques années d'expérience (> 25 ans)
- > MVP Data Platform (12 ans)
- > <http://www.datafly.fr> & <http://blog.datafly.fr>
- > <http://www.mssql.fr>
- > <https://youtube.com/datafly>
- > <https://github.com/datafly/ssmsinforeports>
- > <https://www.linkedin.com/in/arianpapillon/>



# SQL Server : l'optimiseur

- ◎ L'optimiseur décide (plutôt bien) du plan d'exécution
  - > Choix des méthodes d'accès (scan, search)
  - > Choix des types de jointure (par itération, hachage, fusion)
  - > Etc...
- ◎ Cela permet aux plans d'exécution de s'adapter au contexte
  - > Selon les données existantes dans la base
- ◎ Le meilleur choix est généralement de le laisser faire



# Mais s'il se trompe ?

- L'optimiseur peut parfois prendre de mauvaises décisions
  - > Mauvaise estimation de cardinalités :
    - statistiques obsolètes ou distribution de données inégale, requêtes ou prédicats trop complexes
- Le plus souvent, il suffit de lui donner plus (ou de meilleures) informations
  - > Statistiques échantillonnées à 100%, statistiques multi-colonnes ou filtrées
- ... ou de mieux écrire ses requêtes...



# Forçage de l'optimiseur

- Dans certains cas, la solution peut consister à forcer son comportement : directives d'optimiseur (ou indicateurs de requêtes - query hints)
  - > Ces directives (hints) peuvent s'activer au niveau de chaque requête voire (pour certaines au niveau de la base entière).
- Si on force un plan d'exécution, SQL Server perd la capacité de s'adapter au contexte !
  - > Si les données évoluent, les directives peuvent devenir contre-performantes
  - > A réserver aux développeurs ou DBA expérimentés !



# Types d'indicateurs



**OPÉRATIONS DE  
JOINTURES, GROUP  
BY, UNION**



**PARAMÈTRES ET  
CARDINALITÉS,  
RECOMPILATION**



**MÉMOIRE,  
PARALLÉLISME**



**FORÇAGE D'INDEX  
OU DE MODE  
D'ACCÈS**



**ISOLATION &  
VERROUILLAGE**



**ACTIVATION OU  
DÉSACTIVATION DE  
FONCTIONNALITÉS  
DE L'OPTIMISEUR**



**ETC...**



# Indicateurs de requête : syntaxe

- Avec le mot clé **OPTION** à la fin de la requête

```
SELECT ... FROM ...
```

```
OPTION (LOOP JOIN, RECOMPILE)
```

- Autres indicateurs : de tables ou de jointure dans la clause **FROM**

```
SELECT ...
```

```
FROM MaTable WITH (INDEX=2,NOLOCK) ...
```



# Forcer les jointures


- OPTION ({ LOOP | HASH | MERGE } JOIN)


SELECT ... FROM ... OPTION (HASH JOIN)


- Choix forcé pour chaque jointure, précisé dans la clause FROM

> LOOP, HASH, MERGE, REMOTE

> Exemple : FROM TABLE1 INNER HASH JOIN TABLE2 ON ...

  
Boucles imbriquées  
(Jointure interne)  
Coût : 0 %  
0.019s  
6 sur  
6 (100%)

  
Correspondances de hash  
(Jointure interne)  
Coût : 38 %  
0.417s  
121317 sur  
120659 (100%)

  
Jointure de fusion  
(Jointure interne)  
Coût : 12 %  
0.065s  
121317 sur  
121317 (100%)





# Quelques indicateurs

- ◎ Group by
  - > {HASH | ORDER} GROUP
- ◎ Union
  - > {CONCAT | HASH | MERGE} UNION
- ◎ Jointures
  - > FORCE ORDER
  - > { LOOP | MERGE | HASH } JOIN
- ◎ Columnstore index
  - > IGNORE\_NONCLUSTERED\_COLUMNSTORE\_INDEX



# Valeurs de paramètres, recompilation

- > OPTIMIZE FOR ( @variable\_name { UNKNOWN | = <literal\_constant> } [ , ...n ] )
- > OPTIMIZE FOR UNKNOWN
- > RECOMPILE

```
SELECT Nom FROM clients WHERE Nom = @a AND CodePostal = @b OPTION (OPTIMIZE FOR (@a = N'Dupont Jean', @b UNKNOWN))
```

```
SELECT Nom FROM clients WHERE Nom = @a AND CodePostal = @b OPTION (OPTIMIZE FOR UNKNOWN)
```

```
SELECT Nom FROM clients WHERE Nom = @a AND CodePostal = @b OPTION (RECOMPILE)
```



# Autres indicateurs

## ◎ Divers

- > FAST <integer\_value>
- > KEEPFIXED PLAN
- > KEEP PLAN
- > ROBUST PLAN
- > NO\_PERFORMANCE\_SPOOL
- > QUERYTRACEON <integer\_value>
- > MAXRECURSION <integer\_value>



# Parallélisme et mémoire

- > MAXDOP <integer\_value>
- > MAX\_GRANT\_PERCENT = <numeric\_value>
- > MIN\_GRANT\_PERCENT = <numeric\_value>

```
SELECT ... FROM ... OPTION (MAXDOP 2, MAX_GRANT_PERCENT=10)
```



# Indicateurs de tables

- ⦿ Avec le mot clé WITH (...), pour la table dans la requête
  - > L'omission du mot clé WITH est déconseillée et dépréciée

```
SELECT ... FROM ... WITH (NOLOCK, INDEX = 2) WHERE ...
```

- ⦿ Aussi utilisable avec OPTION (TABLE\_HINT ...) à la fin de la requête.

```
SELECT * from clients where nom = @P1  
OPTION (TABLE HINT (clients, FORCESEEK))
```



# Forçage d'index ou méthodes d'accès

## ◉ Index et méthodes d'accès

- > INDEX ( <index\_value> [ , ...n ] ) | INDEX = ( <index\_value> )
- > FORCESEEK [ ( <index\_value> ( <index\_column\_name> [ , ... ] ) ) ]
- > FORCESCAN

```
SELECT ... FROM clients WITH (INDEX (nix$client$nom))  
WHERE ...
```

```
SELECT ... FROM clients WITH (INDEX = 3) WHERE ...
```



# Isolation et verrouillage

- > NOLOCK ou READUNCOMMITTED
- > HOLDLOCK ou SERIALIZABLE
- > READCOMMITTED
- > READCOMMITTEDLOCK
- > REPEATABLEREAD
- > SNAPSHOT
- > NOWAIT, READPAST
- > PAGLOCK, ROWLOCK, TABLOCK, TABLOCKX, UPDLOCK, XLOCK



# Indicateurs de requêtes avec « use hint »

- ⦿ Avec USE HINT (...) dans la clause OPTION
  - > Essentiellement des traceflags

SELECT ...

FROM ...

OPTION (USE HINT ( 'FORCE\_LEGACY\_CARDINALITY\_ESTIMATION' ))





# Indicateurs de requêtes avec « use hint »

- Voir la liste avec `sys.dm_exec_valid_use_hints`
  - > 'ASSUME\_MIN\_SELECTIVITY\_FOR\_FILTER\_ESTIMATES' (TF 4137)
  - > 'ASSUME\_FULL\_INDEPENDENCE\_FOR\_FILTER\_ESTIMATES' (TF 9472)
  - > 'ASSUME\_PARTIAL\_CORRELATION\_FOR\_FILTER\_ESTIMATES'
  - > 'ASSUME\_JOIN\_PREDICATE\_DEPENDS\_ON\_FILTERS' (TF 9476)
  - > 'DISABLE\_BATCH\_MODE\_ADAPTIVE\_JOINS' (SQL 2017+)
  - > 'DISABLE\_BATCH\_MODE\_MEMORY\_GRANT\_FEEDBACK' (SQL 2017+)
  - > 'DISABLE\_DEFERRED\_COMPILATION\_TV' (SQL 2019+)
  - > 'DISABLE\_INTERLEAVED\_EXECUTION\_TVF' (SQL 2017+)
  - > 'DISABLE\_OPTIMIZED\_NESTED\_LOOP' (TF 2340)
  - > 'DISABLE\_OPTIMIZER\_ROWGOAL' (TF 4138)
  - > .../...



# Indicateurs de requêtes avec « use hint »

- > 'DISABLE\_PARAMETER\_SNIFFING' (TF 4136)
- > 'DISABLE\_ROW\_MODE\_MEMORY\_GRANT\_FEEDBACK' (SQL 2019+)
- > 'DISABLE\_TSQL\_SCALAR\_UDF\_INLINING' (SQL 2019+)
- > 'DISALLOW\_BATCH\_MODE' (SQL 2019+)
- > 'ENABLE\_HIST\_AMENDMENT\_FOR\_ASC\_KEYS' (TF 4139)
- > 'ENABLE\_QUERY\_OPTIMIZER\_HOTFIXES' (TF 4199)
- > 'FORCE\_DEFAULT\_CARDINALITY\_ESTIMATION'
- > 'FORCE\_LEGACY\_CARDINALITY\_ESTIMATION' (TF 9481)
- > 'QUERY\_OPTIMIZER\_COMPATIBILITY\_LEVEL\_n' (SQL 2017+)
- > 'QUERY\_PLAN\_PROFILE' (SQL 2016 SP2CU3+)



# USE PLAN

- ◎ Forçage du plan (à partir du plan xml)
  - > USE PLAN N'<xml\_plan>'
- ◎ 

```
SELECT Nom,IdClient FROM Persons WHERE Nom LIKE @a  
OPTION(USE PLAN N'<?xml version="1.0" encoding="utf-16"?>  
<ShowPlanXML xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
Version="1.564" Build="16.0.4035.4"  
xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showpl  
an"> .../... ')
```



# Utilisations des indicateurs

## ⦿ Pour une requête individuelle

```
> SELECT * FROM clients WITH (NOLOCK)  
WHERE Nom = @nom OPTION (OPTIMIZE FOR UNKNOWN)
```

## ⦿ Pour un guide de plan

```
> EXEC sp_create_plan_guide @name = N'MyPlanGuide'  
  , @stmt = N'select * from clients where nom LIKE @P1'  
  , @type = N'SQL'  
  , @module_or_batch = NULL  
  , @params = N'@P1 nvarchar(50)'  
  , @hints = N'OPTION (TABLE HINT (clients, FORCESEEK))'
```

## ⦿ Pour utilisation avec le query store (SQL 2022 !)

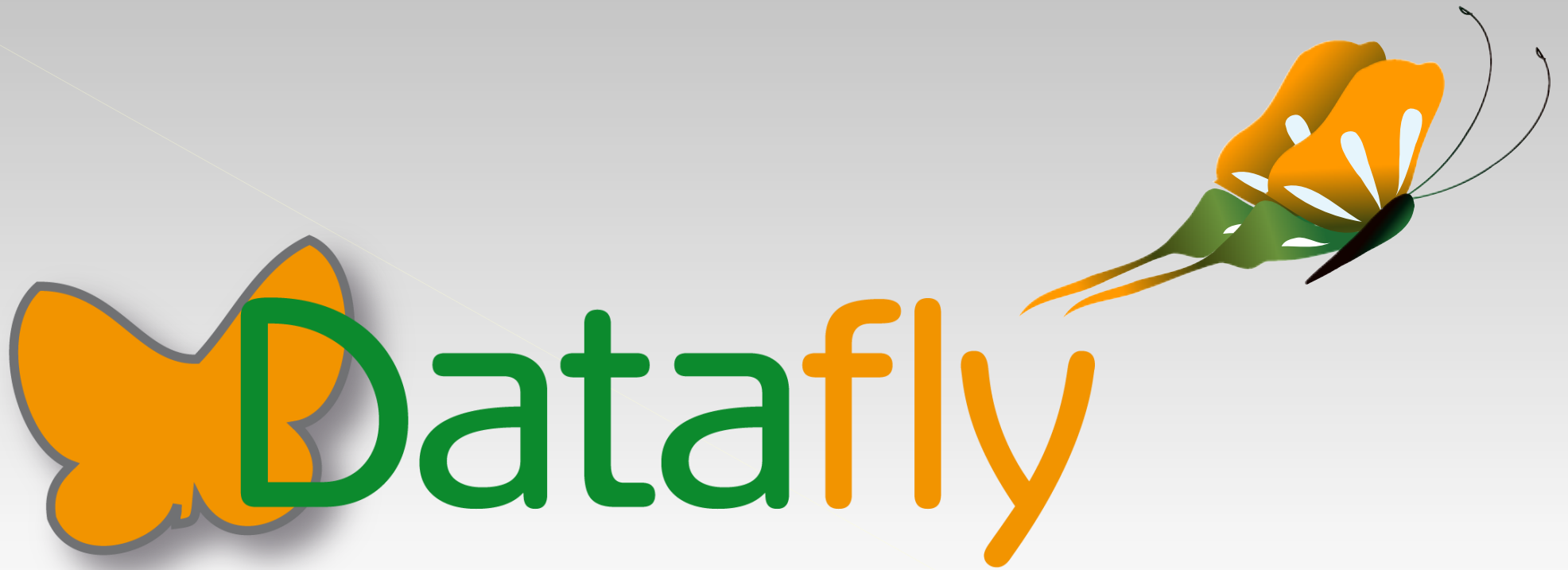
```
> EXEC sp_query_store_set_hints @query_id=5, @value = N'OPTION(RECOMPILE)';
```





#datafogs





<http://www.datafly.fr>

