



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Korszerű adatbázisok előadás 03

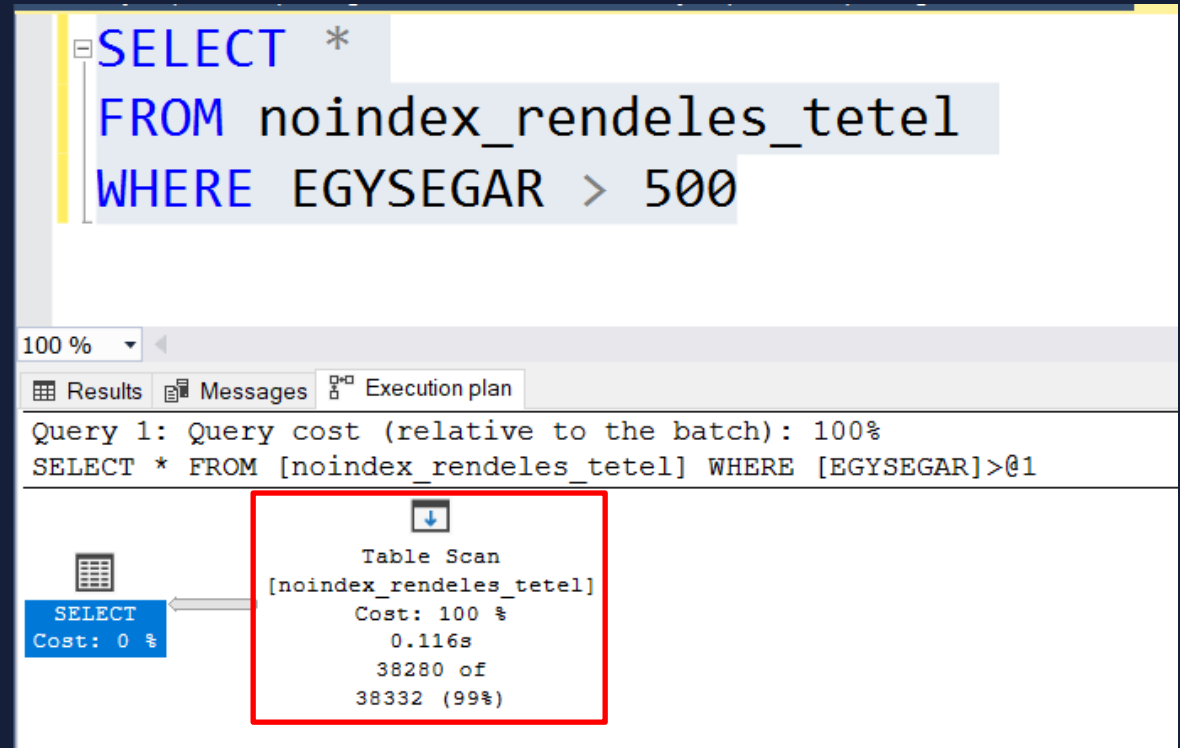
Miről lesz szó?

Lekérdezések optimalizálása

- Indexek
- Join-ok
- Eszközök
 - Lekérdezés végrehajtási terv
 - SQL Server Profiler
 - Tuning Advisor

Hogyan lehet gyorsítani a lekérdezések sebességét?

Sok esetben az a probléma, hogy keresésnél akár az összes rekordot végig kell nézni (TABLE SCAN)



ÖTLET1: Rendezzük fizikailag sorrendbe az adatokat!

Mi a baj a fizikai rendezéssel?

Sok adatot kell mozgatni

Vagy növekvő,
vagy csökkenő

Csak egyféle
rendezési
szempont

DML
utasításoknál
újra kell rendezni

DML = Data Manipulation
Language
(Adatok beszúrása, törlése,
módosítása)

Mi a baj a fizikai rendezéssel?

RAKTAR_KOD	RAKTAR_NEV	RAKTAR_CIM
7	Báthory	1152 Budapest Báthory u. 2.
8	Havanna	1181 Budapest Havanna u. 3
4	KIS	1039 Budapest Madzsar J. u. 23.
1	Kisfaludy	2135 Csörög Kisfaludy u. 55.
5	NAGY	2089 Telki Nefelejcs u. 27
2	Nagyszeben	1182 Budapest Nagyszeben u. 12.
3	Toldy	1015 Budapest Toldy F. u. 58.
6	Veresegyházi	2112 Veresegyház Új iskola u. 10.

INSERT INTO Raktar
VALUES
(9, 'Lapos', '1016
Budapest Krisztina krt')

ÖTLET2: Rendezzük logikailag sorrendbe az adatokat!

Mit jelent a logikai rendezés?

Csak a rendezés alapjául szolgáló mezőt (vagy kifejezést) és a rekordok azonosítóját (mutató, memóriacím) tároljuk

- Az adatok eredeti tárolási sorrendje nem változik
- Egy táblához több logikai rendezést is létrehozhatunk
- A logikai rendezést indexelésnek is nevezik

Az index a táblához vagy nézethez rendelt olyan speciális adatstruktúra, amely felgyorsítja a lekérdezések sebességét.

INDEX (Név szerint)	
Név	ID
Bódi István	D02
Fehér Katalin	D04
Kiss Béla	D01
Nagy Ilona	D03

DOLGOZÓ			
ID	Név	Életkor	...
D01	Kiss Béla	22	
D02	Bódi István	18	
D03	Nagy Ilona	32	
D04	Fehér Katalin	18	

Index - példa

dbo.Rendeles_tetel

- Columns
 - SORSZAM (PK, FK, int, not null)
 - TERMEKKOD (PK, FK, nvarchar(255), not null)
 - EGYSEGAR (float, null)
 - MENNYISEG (float, not null)
- Keys
 - PK_Rendeles_tetel
 - FK_Rendeles_tetel_Rendeles
 - FK_Rendeles_tetel_Termek
- Constraints
- Triggers
- Indexes
 - PK_Rendeles_tetel (Clustered)

```
SELECT *
FROM rendeles_tetel
WHERE EGYSEGAR > 500
```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [rendeles_tetel] WHERE [EGYSEGAR]>@1

SELECT
Cost: 0 %

Clustered Index Scan (Clustered Index on [Rendeles_tetel].[PK_Rendeles_tetel])
Cost: 100 %
0.057s
38280 of 38149 (100%)

A Rendeles_tetel táblához indexet készítve a végrehajtási idő töredékére csökken

Indexek csoportosítása

Egyedi \leftrightarrow Duplikált

- Egy index érték csak egyszer fordulhat-e elő?

Sűrű \leftrightarrow Ritka

- Minden adatrekordhoz készül index bejegyzés?

Egyszerű \leftrightarrow Összetett

- Egy vagy több mezőre épül-e?

Növekvő \leftrightarrow Csökkenő

- Milyen irányú a rendezés?

Sűrű vs. Ritka index

China	• →	China	Beijing	3,705,386
Canada	• →	Canada	Ottawa	3,855,081
Russia	• →	Russia	Moscow	6,592,735
USA	• →	USA	Washington	3,718,691

Az indexmutató egy rekordra mutat (sűrű index).

China	• →	China	Beijing	3,705,386
Russia	• ↘	Canada	Ottawa	3,855,081
USA	• ↘	Russia	Moscow	6,592,735
		USA	Washington	3,718,691

Az indexmutató egy blokkra mutat (ritka index).
A blokkon belül a keresés szekvenciális



Index adatstruktúrák

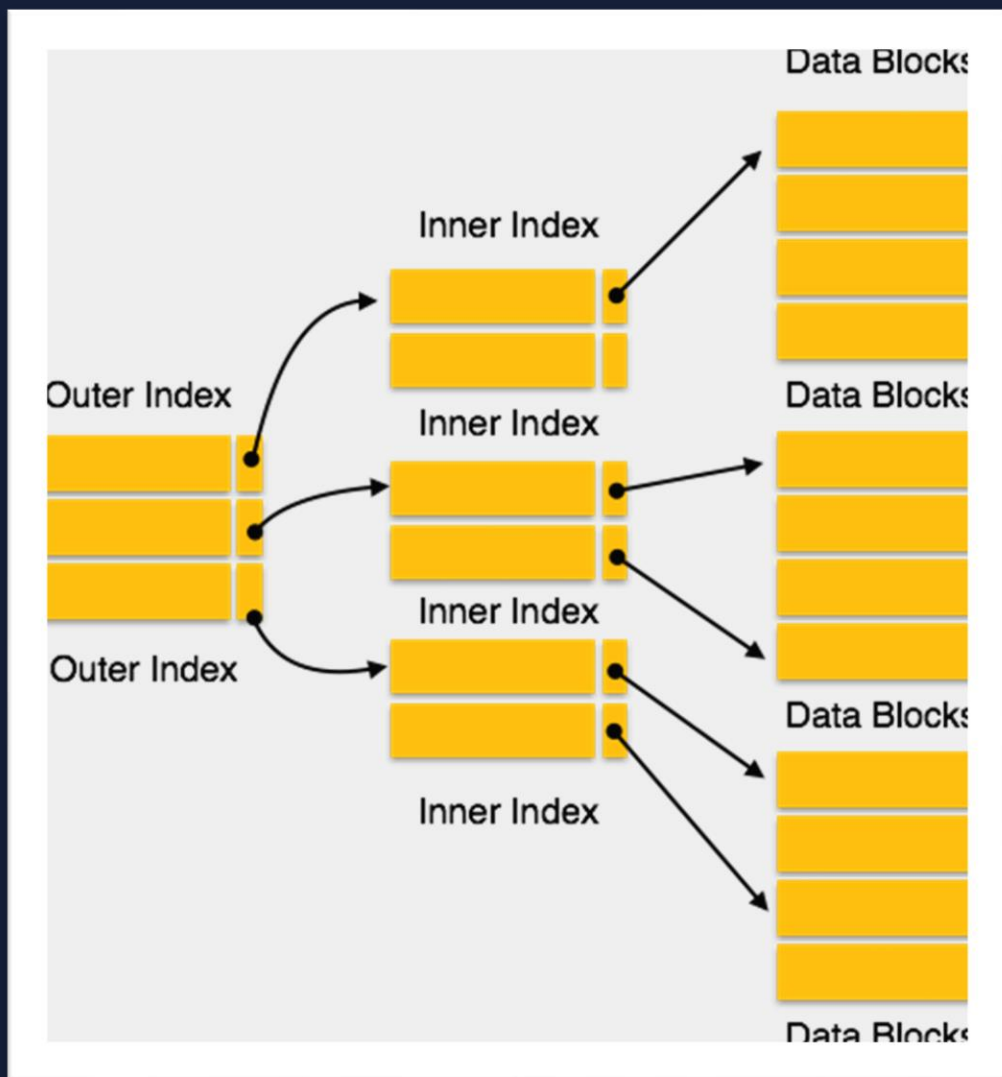
- Egyszintes indexek
- Többszintes indexek
- B-fák
- Hash-alapú indexek
- Bitmap indexek

Egyszintes indexek

Két mezőből álló indextábla, amely az indexelt mező alapján sorba van rendezve. A mutató a rekord fizikai helyére mutat.

Index Tábla		Városok Példa Tábla			
Indexelt mező	Mutató	Város neve	Irányítószám	Megye	Lakosságszám
1234			4567		
2222			8321		
3211			2222		
4567			7789		
5432			1234		
6000			3211		
7789			6000		
8321			5432		

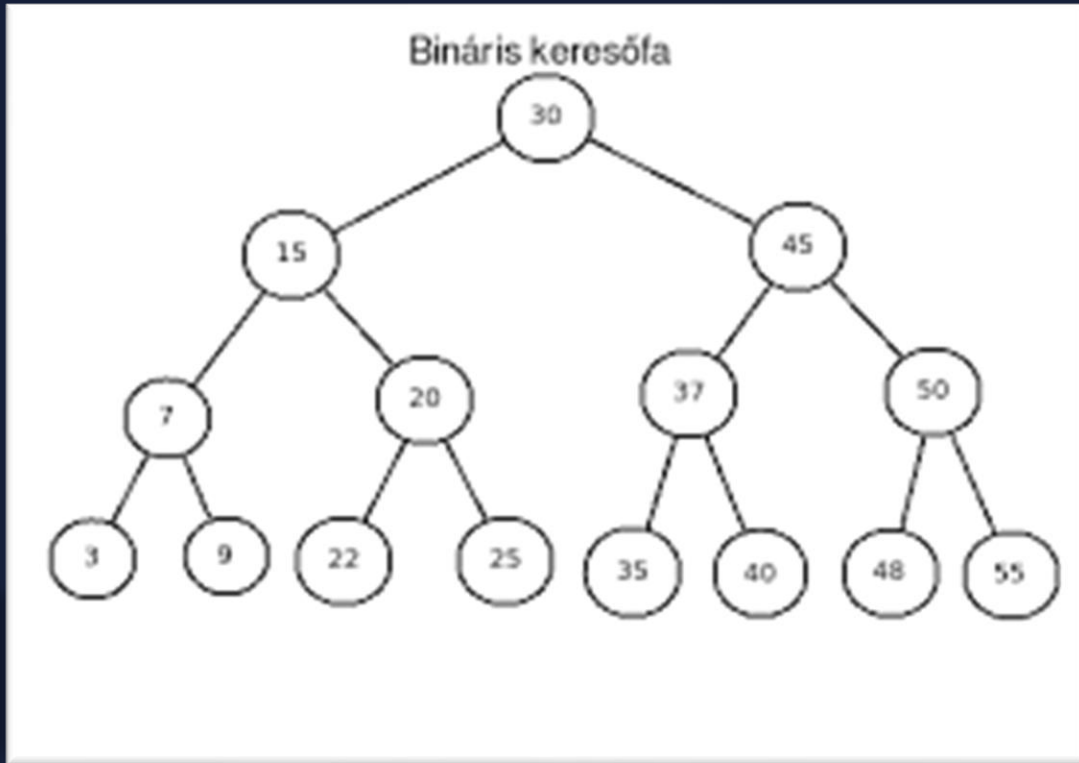
Többszintes indexek



Az indexekhez is indexet készítünk

- Az index így kisebb részekből áll
- Hasznos, ha az egyszintes index nem fér el a memóriában
- Kevesebb blokk olvasás szükséges az adat megtalálásához

Keresőfák

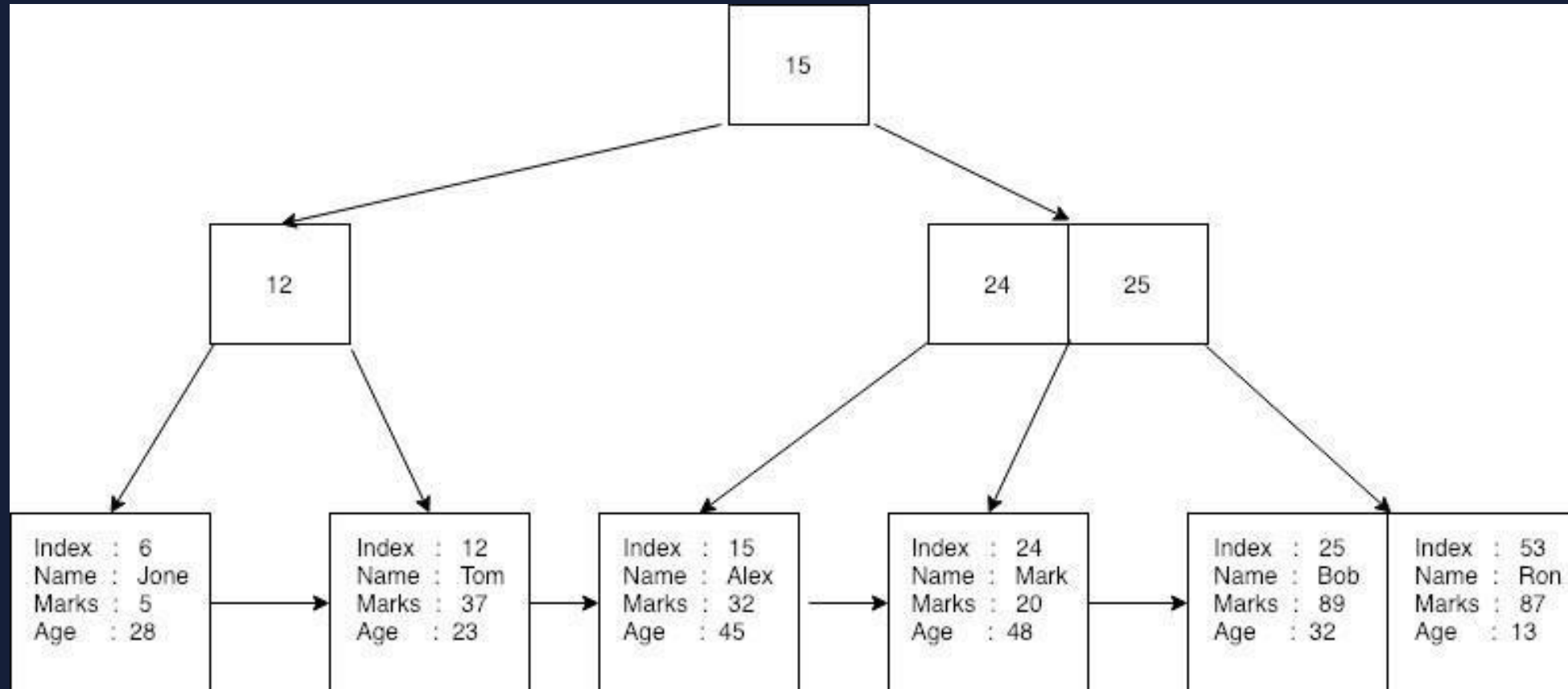


Bármely csomópontból kiindulva a csomópont bal részfájában csak a csomópontban elhelyezetttnél kisebb, a jobb részfájában pedig csak nagyobb értékek szerepelnek

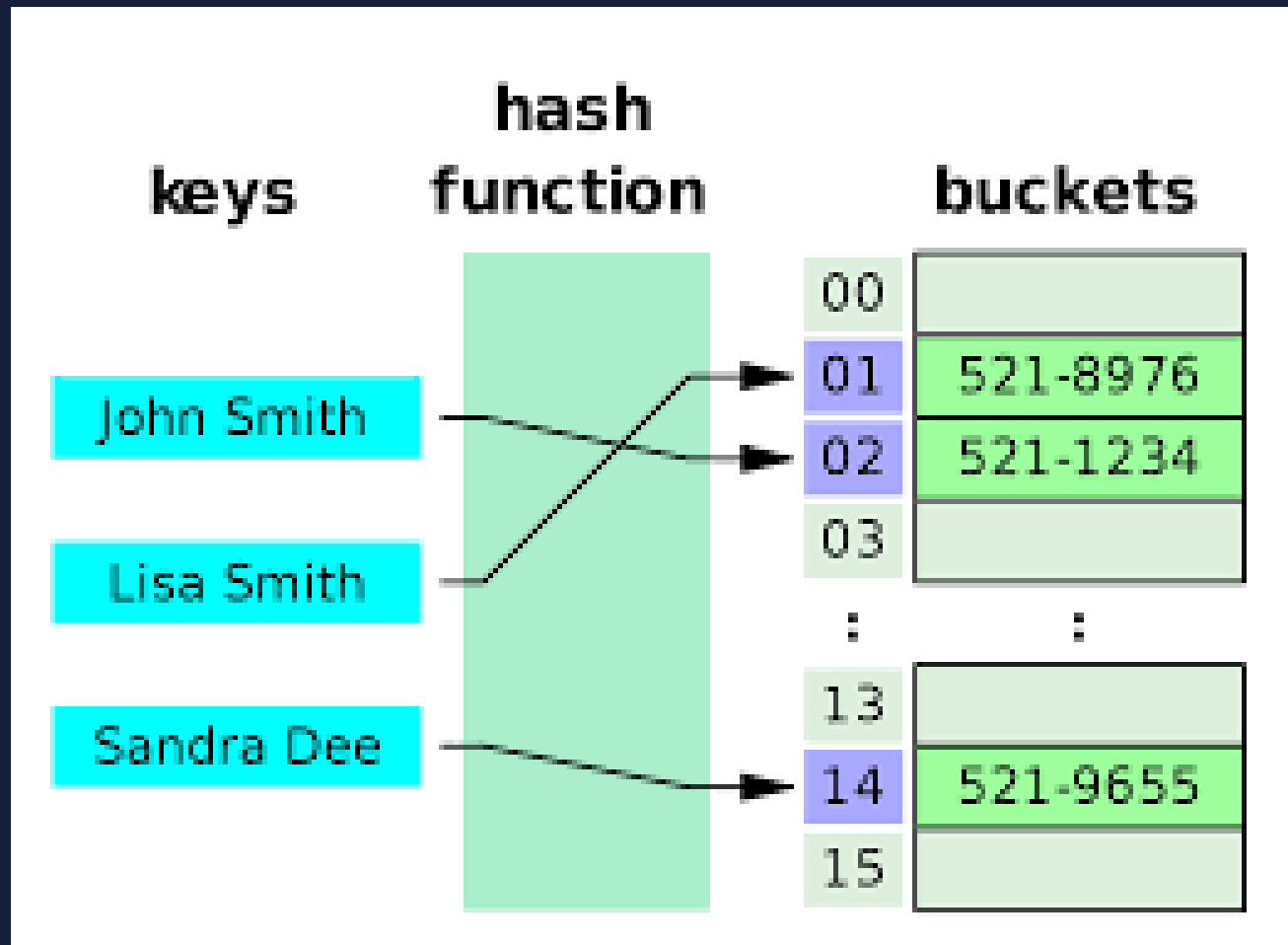
B-fák tulajdonságai

- A gyökértől a levelekig vezető utak hossza egyforma
- Az indexek a B-fa csomópontjaiban helyezkednek el
- Az adatok helyét jelző mutató csak a levelekben található
- A struktúra lehetővé teszi a soros és a random elérést is

B-fák (kiegyensúlyozott keresőfák)

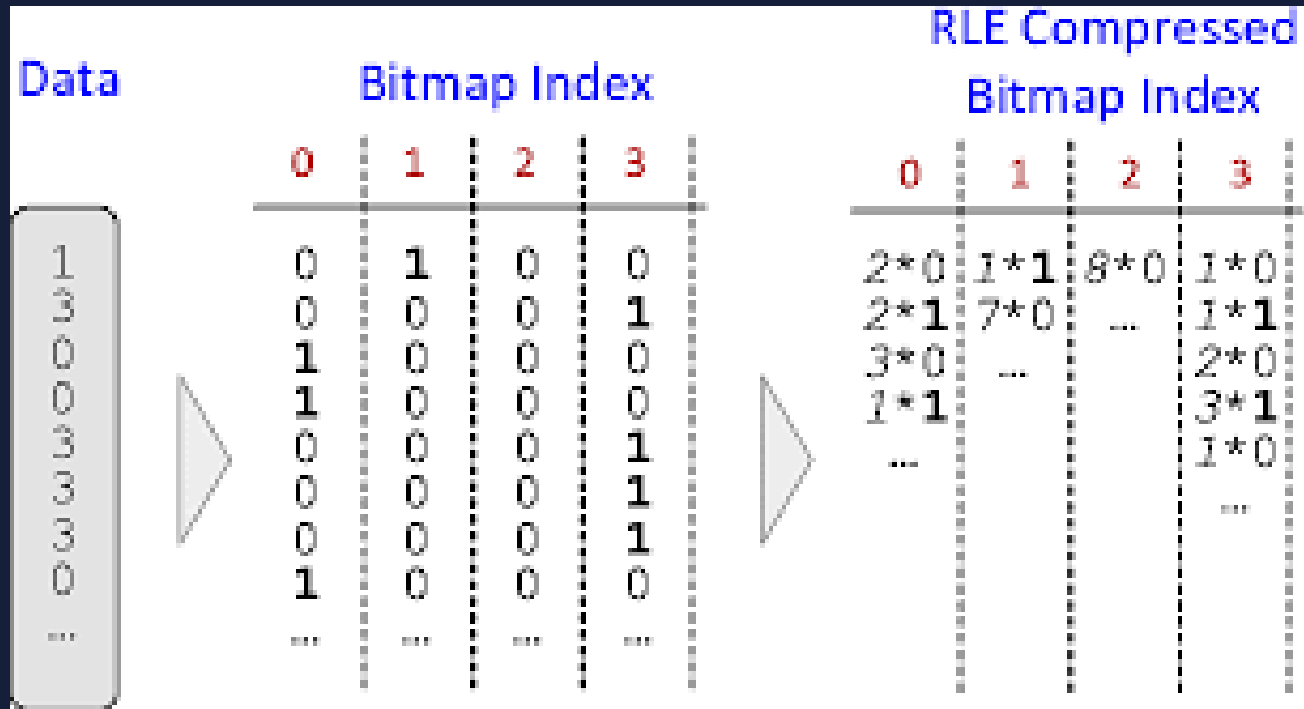


Hash-alapú indexek



- Az adatok csoportokba vannak rendezve
- A hash függvény adja meg, hogy melyik csoportban van az adat

Bitmap indexek



- Olyan oszlopokra alkalmazzuk, ahol kevés az egyedi érték
- Az index tömöríthető is



Fontosabb T-SQL Index típusok

- Clustered
- Non-clustered
- Columnstore
- XML
- Spatial

Az indexek létrejöhetnek automatikusan vagy manuálisan
(CREATE INDEX)

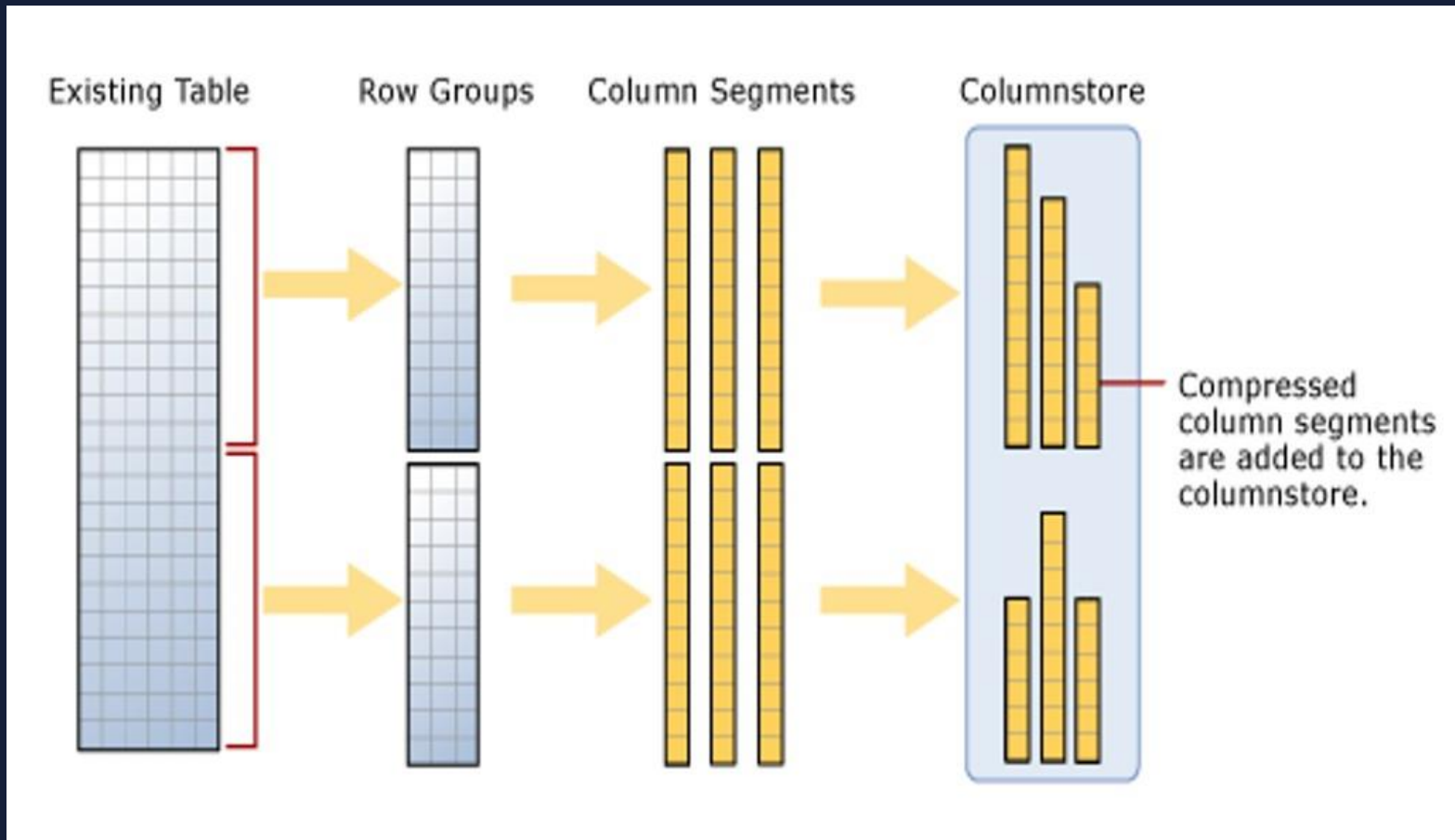
Clustered index

- Az adatokat az index kulcsnak megfelelő sorrendbe rendezi és tárolja.
- A clustered index B-fa struktúrát használ
- Egy tábla esetén csak egy clustered index hozható létre
- Alapértelmezés szerint az elsődleges kulcs definiálásakor automatikusan létrejön

Non-clustered index

- Kulcs-mutató érték párokat tárol.
- Az index sorai a kulcs értékeknek megfelelő sorrendben vannak tárolva
- Az adatok tárolási sorrendje ettől eltérő
- Egy tábla esetén több non-clustered index is létrehozható
- Speciális lehetőségek: Unique, Filtered, Included columns

Columnstore index



- A columnstore index lehet
 - Clustered
 - Non-clustered
- Egy táblához csak egy columnstore index készíthető

Adattárházakból való lekérdezéseknél kiemelten fontos!

Oszlop-alapú tárolás és lekérdezés végrehajtás



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Indexek - SSMS

SQLQuery13.sql - 192.168.85.29.DM_STAR (GROUPIT\51282 (174))* - Microsoft SQL Server Management Studio

File Edit View Project Tools Window Help

DM_STAR Execute

Object Explorer

- proba
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - dbo.MDVD
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes**
 - PK_MDVD_CAD94A09AAC07AE0 (Clustered)
 - UQ_MDVD_9C2964183D2236B1 (Unique, Nonclustered)
 - Statistics
 - dbo.MKOLCSOZES
 - dbo.MNEM
 - dbo.MSZEMELY
- Views
- External Resources
- Synonyms

New Index

Ready

Select a page

- General
- Options
- Storage
- Filter
- Extended Properties

Script Help

Table name: MDVD

Index name: NonClusteredIndex-20200309-222902

Index type: Nonclustered

☐ Unique

Index key columns Included columns

Name	Sort Order	Data Type	Size	Identity	Allow NULLs	
SORSZAM	Ascending	numeric(4,1)	5	No	No	Add... Remove
FILM	Ascending	varchar(12)	128	No	No	Move Up Move Down

Connection

.sqlexpress [GROUPIT\51282]

View connection properties

Progress

Ready

Index létrehozása parancssal – példa1

CREATE [[**UNIQUE**][**CLUSTERED**][**COLUMNSTORE**]] **INDEX** indexnév
ON táblanév (oszlop1 [ASC] | [DESC], oszlop2 [ASC] | [DESC] ...) [INCLUDE (oszlopok)]

SQLQuery8.sql - hal...hop (hallgato (87))* SQLQuery5.sql - hal...hop (hallgato (93))* SQLQuery3.sql - hal...hop (hallgato (70))

```
-- Nonclustered Unique index a Raktár táblára
-- raktárkód szerint növekvő

CREATE UNIQUE INDEX I_Raktar ON Raktar
(
    RAKTAR_KOD ASC
)
```

UNIQUE kényszer létrehozásakor a megfelelő UNIQUE index automatikusan létrejön!

Index létrehozása parancssal – példa2

CREATE [[UNIQUE][**CLUSTERED**][COLUMNSTORE]] **INDEX** indexnév
ON táblanév (oszlop1 [ASC] | [DESC], oszlop2 [ASC] | [DESC] ...)) [INCLUDE (oszlopok)]

```
SQLQuery11.sql - h...bshop (ujaenb (89))*  SQLQuery8.sql - hal...hop (hallgato (87))*  SQLQuery5.sql - hal...hop (hallgato (93))*  SQLQuery
-- Clustered index a noindex_rendeles táblára
-- sorszam szerint növekvő

CREATE CLUSTERED INDEX I_rendeles ON noindex_rendeles
(
    SORSZAM ASC
)
```

Elsődleges kulcs létrehozásakor a Clustered Index automatikusan létrejön!

Index létrehozása parancssal – példa3

CREATE [[UNIQUE][CLUSTERED][COLUMNSTORE]] **INDEX** indexnév
ON táblanév (oszlop1 [ASC] | [DESC], oszlop2 [ASC] | [DESC] ...)) [INCLUDE (oszlopok)]

SQLQuery13.sql - ...BKAE\gmolnar (61))* SQLQuery12.sql - h...bshop (ujaenb (89))* SQLQuery2.sql - hal...hop (hallgato (58))*

```
--COLUMNSTORE Index létrehozása az Ugyfel  
--tábla LOGIN és EMAIL oszlopaival  
  
CREATE COLUMNSTORE INDEX i_login_email_ugyfel  
ON Ugyfel (LOGIN, EMAIL)
```

A Clustered index létrehozását az Azure SQL nem minden szolgáltatásrétege támogatja

Melyik index legyen használva?

HINT- Lekérdezési tipp

Használata:

SELECT

FROM...

...

ORDER BY ...

OPTION (TABLE HINT (táblanév, INDEX(indexnév)))

- Többféle HINT létezik:
 - JOIN HINT
 - TABLE HINT
 - QUERY HINT
- Céljuk:
 - Preferált vagy tiltott index
 - JOIN típus, sorrend
 - Táblaelérési mód
 - Párhuzamos végrehajtás

HINT példa

Object Explorer

- Database Diagrams
- Tables
 - System Tables
 - External Tables
 - GraphTables
 - dbo.noindex_rendeles_tetel
 - dbo.Raktar
 - Columns
 - RAKTAR_KOD (PK, int, not null)
 - RAKTAR_NEV (nvarchar(255), not null)
 - RAKTAR_CIM (nvarchar(255), not null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - IX_Raktar (Unique, Non-Clustered)
 - PK_Raktar (Clustered)
 - Statistics
 - dbo.Rendeles
 - dbo.Rendeles_tetel
 - dbo.Termek
 - dbo.Termekategoria
 - dbo.Ugyfel
 - Columns
 - Keys

SQLQuery7.sql - hal...hop (hallgato (55))* SQLQuery6.sql - hal...hop (hallgato (53))*

```

SELECT *
FROM Raktar
WHERE RAKTAR_NEV LIKE '%a%'
OPTION (TABLE HINT (Raktar, INDEX (IX_Raktar)))
    
```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT * FROM Raktar WHERE RAKTAR_NEV LIKE '%a%' OPTION (TABLE HINT (Raktar, INDEX (IX_Raktar)))

```

graph TD
    A[SELECT  
Cost: 0 %] --> B[Nested Loops  
(Inner Join)  
Cost: 0 %  
0.000s  
5 of 5 (100%)]
    B --> C[Index Scan (NonClustered)  
[Raktar].[IX_Raktar]  
Cost: 42 %  
0.000s  
9 of 9 (100%)]
    B --> D[Key Lookup (Clustered)  
[Raktar].[PK_Raktar]  
Cost: 58 %  
0.000s  
5 of 45 (11%)]
    
```

Indexek hátrányai

- Tárhelyet foglalnak
- Folyamatos karbantartást igényelnek
 - Minden DML-művelet esetén (automatikus)
 - Újraépítés (Rebuild) – törli és újból létrehozza az indexet
 - Újraszervezés (Reorganize) – töredezettségmentesítést hajt végre
 - Tömörítés
 - Index statisztikák – a lekérdezések optimalizálását segítik
- A DML-műveleteket lelassítják
- Nem ajánlottak
 - Kis táblák esetén
 - Sok NULL értéket tartalmazó oszlopra
 - Olyan oszlopokra, amelynek értékei gyakran változnak



Táblák összekapcsolása

- Akkor lehet rá szükség, ha a keresett információ egynél több táblában található meg
- **A kapcsolat alapja legtöbbször az egyik tábla idegen kulcsának és a másik tábla kulcsának azonos értéke**, ritkább esetben a két tábla egy-egy oszlopára vonatkozó összehasonlító feltétel
- Az összekapcsolás megvalósításának tipikus módja:

SELECT oszlopnév_lista

FROM táblanév1 aliasnév1 **INNER JOIN** táblanév2 aliasnév2

ON aliasnév1.idegenkulcsérték = aliasnév2.kulcsérték

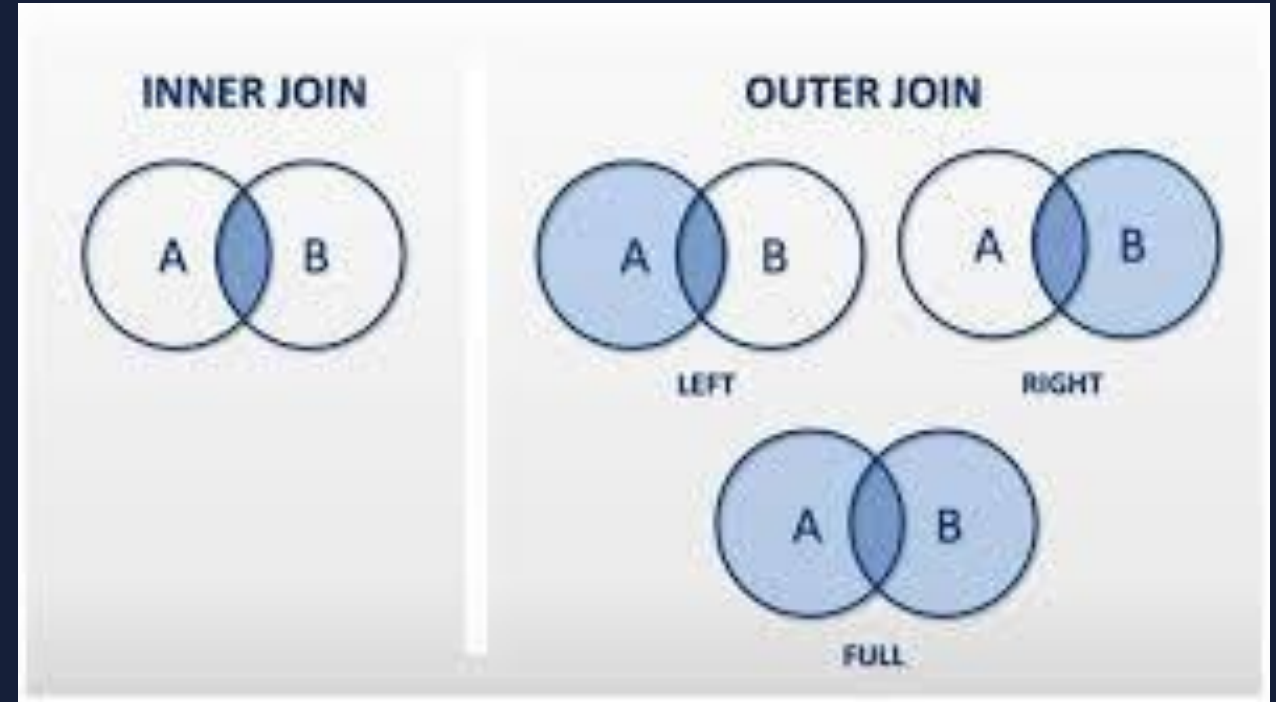
Táblák összekapcsolása - Megjegyzések

- Az aliasnevek (másodlagos táblanevek, rövid nevek) használata javítja a lekérdezés olvashatóságát*
- Egy tábla akár saját magával is összekapcsolható, ilyenkor az aliasnevek használata kötelező
- Mivel a két táblának lehetnek azonos nevű mezői is, ezért a SELECT utáni oszlopnév listában használjunk minősített oszlopneveket (tábla aliasnév.oszlopnév) vagy aliasnevek hiányában táblanév.oszlopnév formában
- Az INNER kulcsszó használata nem kötelező
- Az ON kulcsszó után összetett feltétel is szerepelhet
- **Az eredménylistából INNER JOIN esetén kimaradnak azok a sorok, ahol az idegenkulcs = kulcs feltétel valamelyik oldalán NULL érték található, pl: a diák azonosítója nincs megadva a munka táblában**

*Egy tábla neve a legáltalánosabb esetben [Szervernév].[Adatbázisnév].[Séma név].[Táblanév] formátumú, nagyon hosszú is lehet

Fontosabb JOIN típusok

- (INNER) JOIN:
Az A tábla idegen kulcsa megegyezik a B tábla kulcsával
- LEFT (OUTER) JOIN:
Az INNER JOIN eredményéhez hozzá veszi az A tábla minden további sorát is
- RIGHT (OUTER) JOIN:
Az INNER JOIN eredményéhez hozzá veszi az B tábla minden további sorát is
- FULL (OUTER) JOINT:
Az INNER JOIN eredményéhez hozzá veszi az A és B tábla minden további sorait is



Az OUTER szó használata nem kötelező

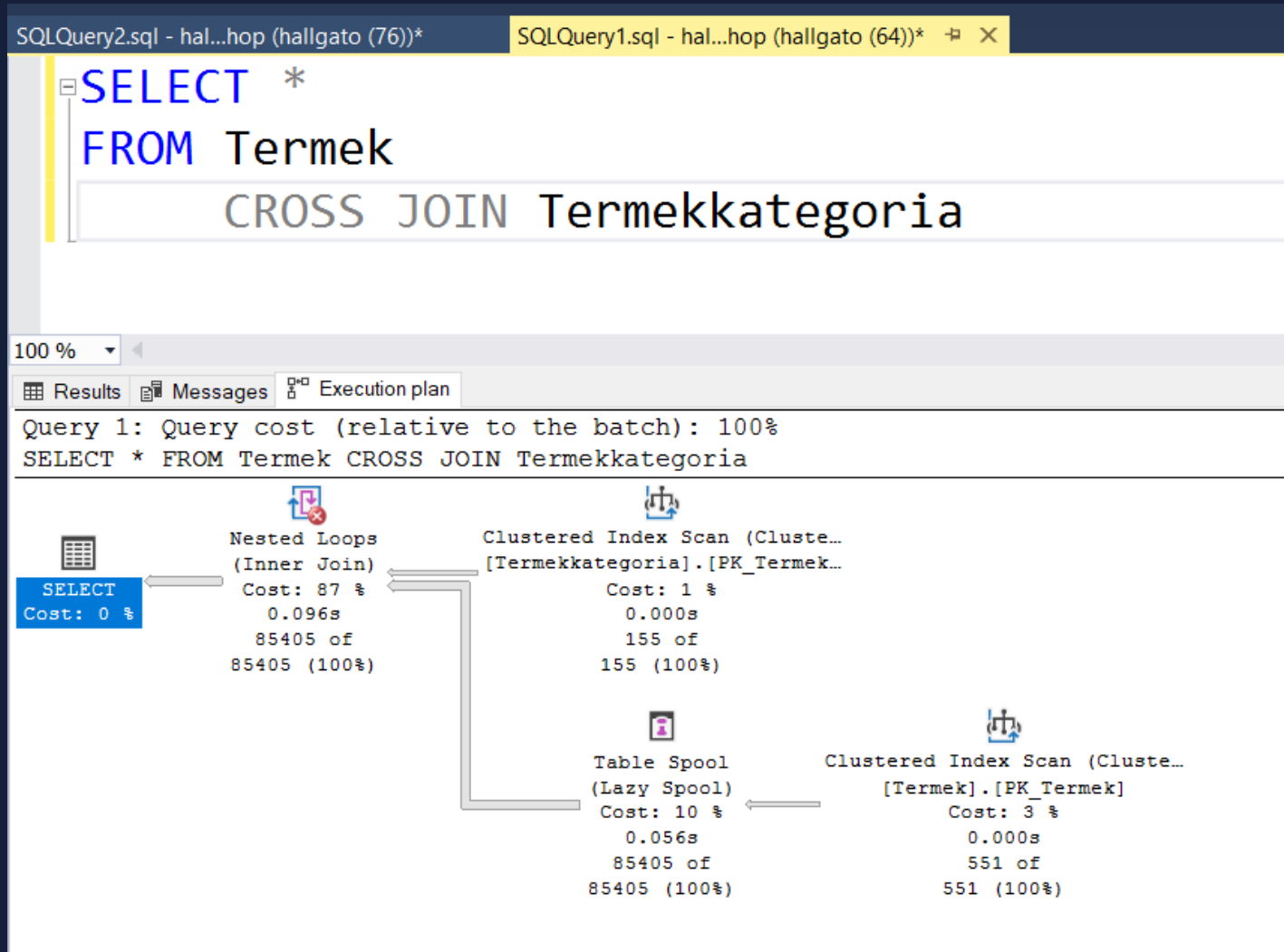
Az első három esetben különbség akkor van, ha az idegenkulcs = kulcs feltétel egyik oldalán NULL érték áll

Az első tábla minden egyes sorát összepárosítja a második tábla minden egyes sorával (Descartes-szorzat)

- Kétféle szintaktika
 - SELECT ...
FROM Tábla1 CROSS JOIN Tábla2
 - SELECT...
FROM Tábla1, Tábla2

A CROSS JOIN-os lekérdezések nagyon költségesek lehetnek!

CROSS JOIN - Példa



Több, mint 85 ezer sor
lesz az eredmény!

(INNER JOIN esetén csak
551 keletkezik)

A táblák összekapcsolását finomhangolják

- Fontosabb HINT-ek

- MERGE
- HASH
- LOOP*
- REMOTE**

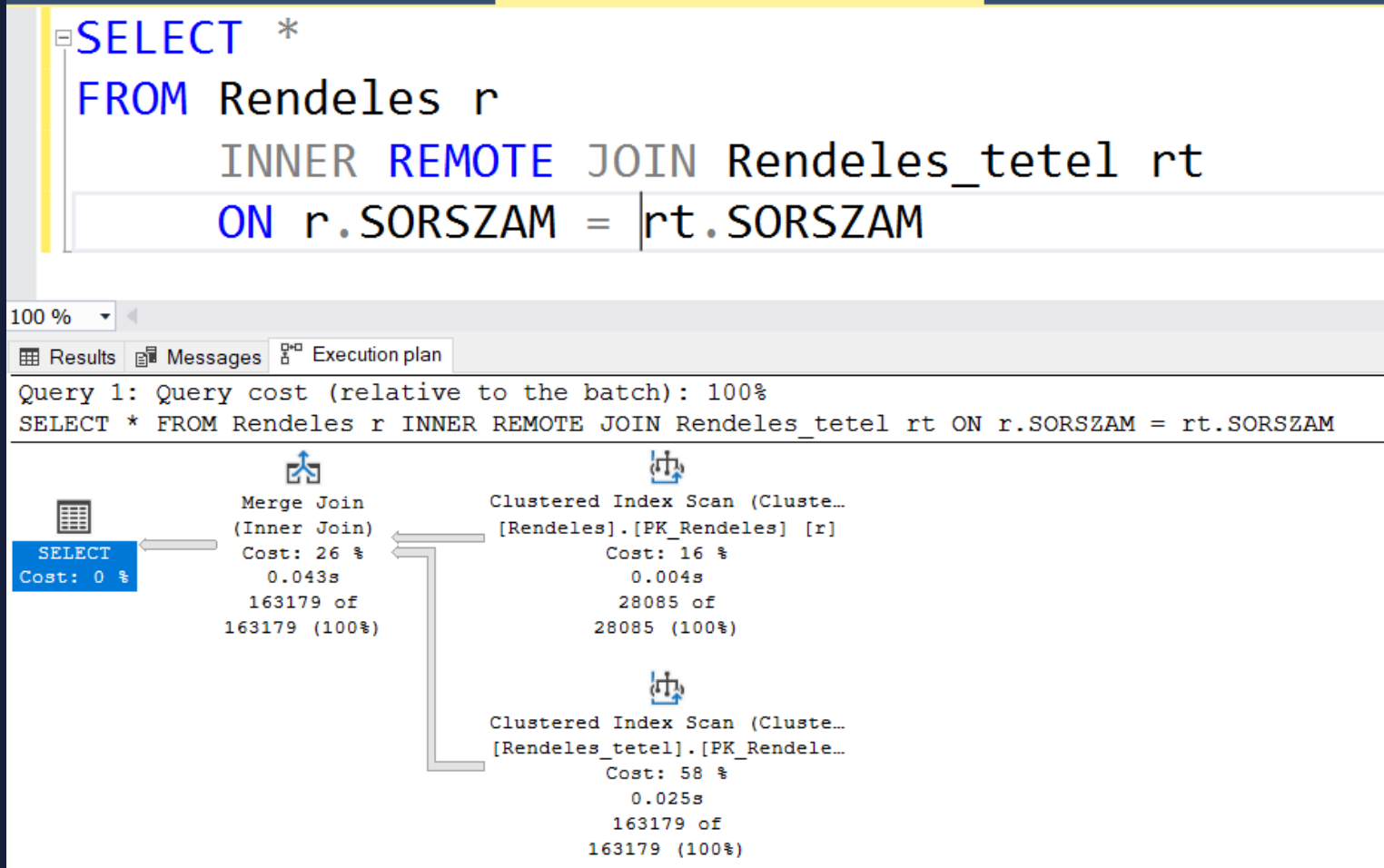
Formája

```
SELECT *
FROM Tábla1 t1
[INNER|OUTER|FULL] [HINT] JOIN Tábla2 t2
ON ...
```

* A LOOP HINT nem használható RIGHT és FULL JOIN esetén

**A REMOTE HINT csak INNER JOIN esetén használható, nem minden verzió támogatja

JOIN HINT - példa



A REMOTE HINT-et akkor érdemes használni, ha

- A jobboldali tábla egy távoli adatforrásban van

ÉS

- Nagyobb méretű, mint a baloldali (helyi) tábla

A JOIN alternatívája, táblaértékű kifejezéseket kapcsol össze

- Két fajtája:
 - CROSS APPLY: a baloldali tábla kifejezésből azokat a sorokat adja vissza, amelyekkel van egyező a jobboldali tábla kifejezésben (~ INNER JOIN)
 - OUTER APPLY: a baloldali tábla kifejezés minden sorát visszaadja, azoknál a soroknál, ahol nincs megfelelő a jobboldali tábla kifejezésben, ott a jobboldali értékeknél NULL értékeket ad vissza (~ LEFT JOIN)

APPLY formája

SELECT ...

FROM tábla1 t1

CROSS | OUTER APPLY

(

SELECT ...

FROM tábla2 t2

WHERE t1.oszlop1 = t2.oszlop2

) AS t

Az APPLY operátorok használata esetén a lekérdezés végrehajtási terve más lesz, mint JOIN-ok esetén.



APPLY vs. JOIN

- Az APPLY operátor esetén a kapcsolat jobb oldala egy tábla értékű függvény is lehet
- Bizonyos esetekben az APPLY operátor használata performanciálisan kedvezőbb lehet a JOIN-nál
- Az APPLY T-SQL specifikus operátor, a JOIN viszont szabványos

APPLY példák

A jobboldalon táblaértékű fv. Is állhat

SQLQuery6.sql - hal...hop (hallgato (54))* x SQLQuery5.sql - not connected* SQLQuery4.s

```

SELECT r.rend_datum,
       u2.nev,
       r.fiz_mod
FROM Rendeles r
CROSS APPLY
(
  SELECT LOGIN, nev
  FROM Ugyfel u
  WHERE r.LOGIN = u.LOGIN
) AS u2
  
```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT r.rend_datum, u2.nev, r.fiz_mod FROM Rendeles r CROSS APPLY
Missing Index (Impact 97.5938): CREATE NONCLUSTERED INDEX [<Name of

```

SELECT r.rend_datum,
       u2.nev,
       r.fiz_mod
FROM Rendeles r
OUTER APPLY
  dbo.GetAllClient() AS u2
WHERE YEAR(r.rend_datum) > 2016
  
```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT r.rend_datum, u2.nev, r.fiz_mod FROM Rendeles r OUTER APPLY dbo.GetAllClient

Lekérdezés optimalizáló - automatikus

Cél a lekérdezés által használt hardver erőforrásigény (CPU, Memória, I/O) csökkentése

Feladatai:

- Fizikai terv elkészítése
- Táblák bejárása
- Táblák összekapcsolása
- Logikai terv
- Optimalizálás (költségbecslés alapján)

Gyakran alkalmazott technikák

- Statisztikák
- Táblaelérési módok
- Indexek
- Hint-ek
- Plan cache – itt tárolja el azokat a végrehajtási terveket, amelyek számításba jöhetnek

Lekérdezés optimalizálás manuálisan

- Egy lekérdezés általában sokféle módon végrehajtható
- A beépített lekérdezés optimalizáló (SQL Query Optimizer) nem mindig találja meg az optimális módot
 - Ez főleg komplex lekérdezéseknél fordulhat elő
 - Ilyenkor nem vizsgál meg minden lehetséges esetet, mert akkor túl sokáig tartana az optimalizálási folyamat
 - Elsősorban azt próbálja megbecsülni, hogy az adott terv végrehajtása hány sort érinthet
 - Gyakori futtatás esetén sokszor a már meglévő tervet használja

Nem megfelelő performancia esetén érdemes felülvizsgálni a végrehajtási tervet

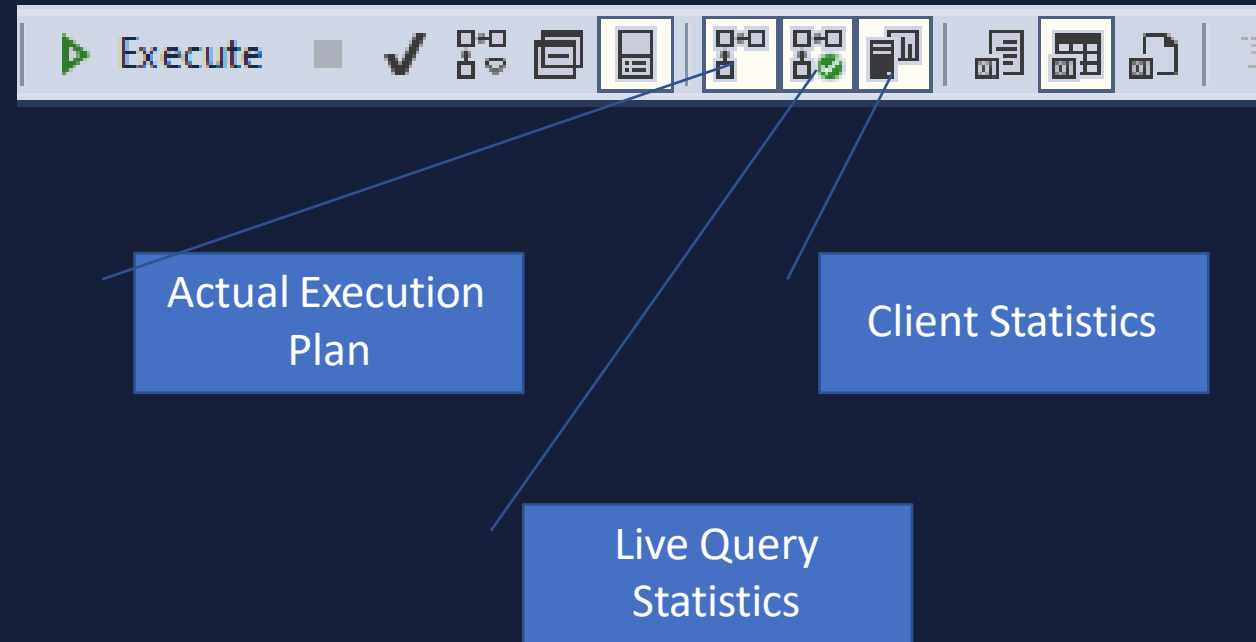
Lekérdezések optimalizálása – néhány javaslat

- SELECT * helyett jobb a SELECT oszloplista
- Lehetőleg kerüljük a DISTINCT használatát
- Helyettesítő karaktereket lehetőleg csak a keresett kifejezés végén használjunk
- A HAVING részben csak aggregált adatot szűrjünk
- Beágyazott SELECT-nél az IN helyett inkább EXISTS operátort használjunk
- UNION helyett nagy tábláknál inkább UNION ALL-t használjunk
- Ha a WHERE feltételben függvényt használunk, akkor az lehetőleg az operátor jobb oldalán legyen

Lekérdezések optimalizálása – néhány javaslat

- A JOIN és a WHERE részekben kerüljük a számított mezők használatát
- Részesítsük előnyben az analitikus függvények használatát
- A GROUP BY oszloplista sorrendjét – ha lehet - úgy válasszuk meg, hogy az elején egyedi értékeket tartalmazó oszlopok legyenek
- A hosszú táblanevek helyett használjunk rövid alias neveket
- A WHERE részben a > jel helyett általában előnyösebb a >=
- Ha a WHERE részben több feltétel van, akkor azt vegyül előre, amely ahhoz a táblához tartozik, amelytől a legtöbb más tábla függ
- A FROM részben pont fordítva, itt a legtöbb függéssel rendelkező tábla legyen az utolsó

- Lekérdezés végrehajtási terv
- SQL Tuning Advisor
- SQL Server Extended Events
- SQL Profiler, XEvent Profiler
- Egyéb
 - SET STATISTICS IO ON/OFF, SET STATISTICS TIME ON/OFF
 - SET SHOWPLAN_TEXT ON/OFF, SET SHOWPLAN_ALL ON/OFF
 - Dynamics Management Objects

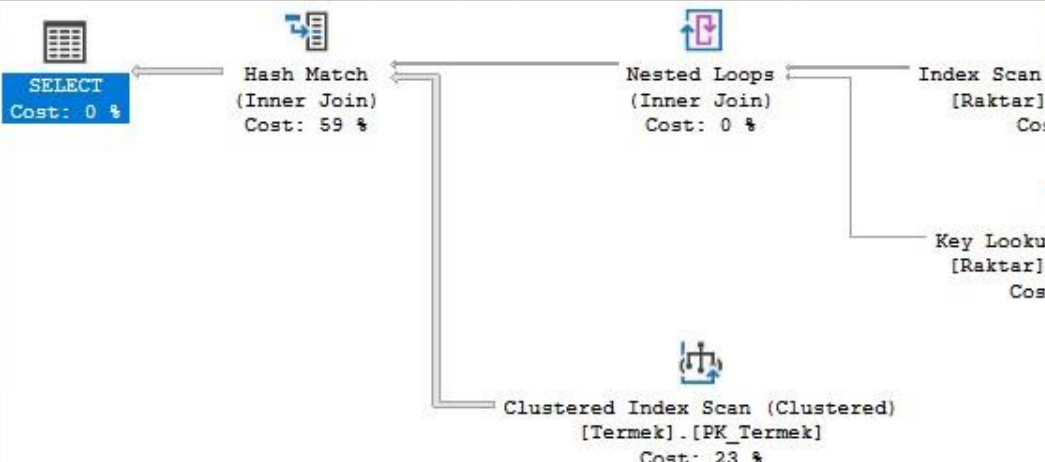


Lekérdezés végrehajtási terv

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
select * from raktar --join termek on raktar.raktar_kod = termek.raktar_kod join ( select * from termek ) t on raktar.raktar_kod = termek.raktar_kod
```



Index Scan (NonClustered)

Scan a nonclustered index, entirely or only a range.

Physical Operation	Index Scan
Logical Operation	Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated I/O Cost	0,003125
Estimated Operator Cost	0,0032919 (8%)
Estimated CPU Cost	0,0001669
Estimated Subtree Cost	0,0032919
Estimated Number of Executions	1
Estimated Number of Rows	9
Estimated Number of Rows to be Read	9
Estimated Row Size	11 B
Ordered	False
Node ID	2

Object
[webshop2].[dbo].[Raktar].[IX_Raktar]















Output List
[webshop2].[dbo].[Raktar].RAKTAR_KOD

Query executed successfully.

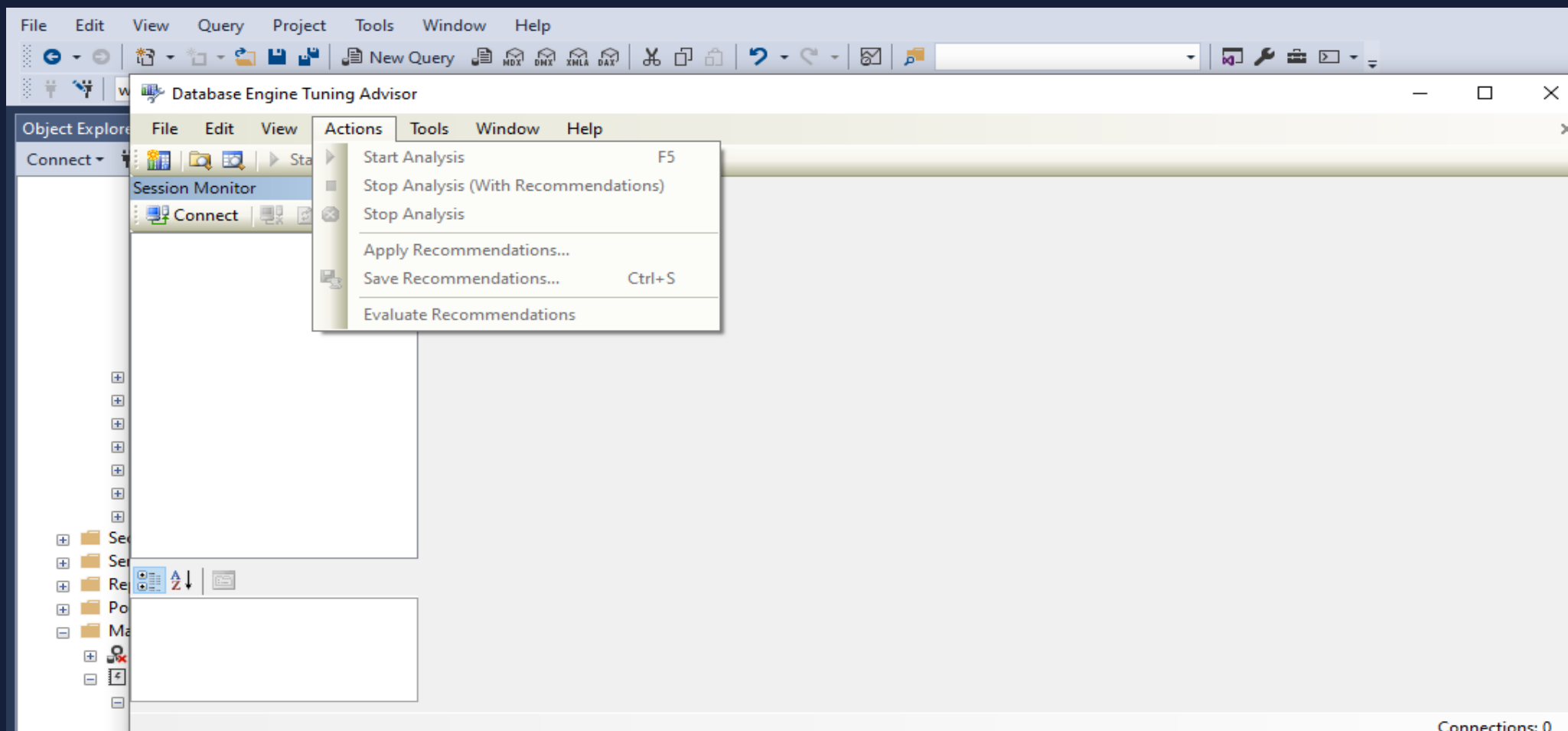
Ln 6 Col 10 Ch 10

SSMS-ben: Query\Display Estimated Execution Plan

Lekérdezés végrehajtási terv – gyakori operátorok

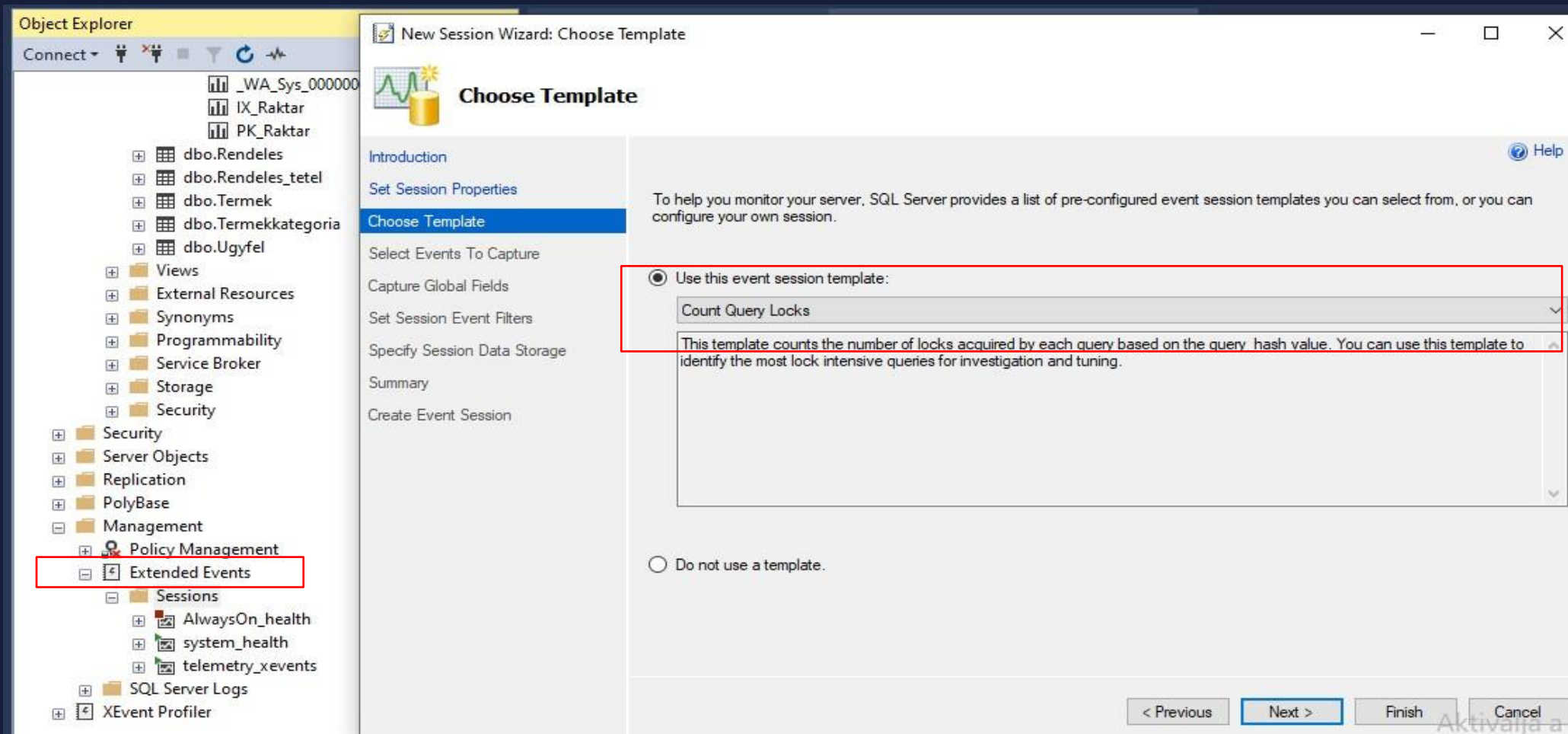
	Table Scan	Scan of a whole table stored as a heap. A table can be organized as a heap or as a clustered index.
	Clustered Index Scan	Scan of a whole table stored as a clustered index. Indexes are stored as balanced trees.
	Clustered Index Seek	SQL Server seeks for the first value in the seek argument (for example, a column value in the WHERE clause) in a clustered index and then performs a partial scan.
	Index Scan	Scan of a whole nonduplicated index.
	Index Seek	SQL Server seeks for the first value in the seek argument (for example, a column value in the WHERE clause) in a nonduplicated index and then performs a partial scan.
	RID Lookup	Lookup for a single row in a table stored as a heap by using its row identifier (RID).
	Key Lookup	Lookup for a single row in a table stored as a clustered index by using the key of the index.
	Hash Match Join	Joins that use the Hash algorithm.
	Merge Join	Joins that use the Merge algorithm.
	Nested Loops	Joins that use the Nested Loops algorithm.
	Stream Aggregate	Aggregation of ordered rows.
	Hash Match Aggregate	Hash algorithm used for aggregating. Note that the icon is the same as the icon for the Hash Match Join; however, in an execution plan, text below the icons gives you information about whether the operator performed a join or an aggregate.
	Filter	Filters rows based on a predicate (for example, a predicate of the WHERE clause).
	Sort	Sort of incoming rows.

Tuning Advisor



Az SQL Server Express és az Azure SQL ezt a funkciót nem támogatja.
Opcionálisan más verzió vagy az Azure Portál teljesítménykezelő funkciói használhatók.

SQL Server Extended Events



Az Extended Events futtatásához ALTER ANY EVENT SESSION jogosultság szükséges.



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

SQL Server XEvent Profiler

Object Explorer

Connect

- haladosql.database.windows.net (SQL Server 12.0.2000)
- haladosql.database.windows.net (SQL Server 12.0.2000)
- SZA0030470\SQLEXPRESS01 (SQL Server 15.0.2000 - E

Databases

- System Databases
- Database Snapshots
- dvd
- SD_LOCAL
- sdcopy
- webshop

Security

- Server Objects
- Replication
- PolyBase

Management

- Policy Management
- Extended Events
 - Sessions
 - SQL Server Logs

XEvent Profiler

- Standard
 - Stop Session
 - Launch Session
 - Reports
 - Refresh
- TSQ

SZA0030470\SQLEXP...andard: Live Data

gyak3_melleklet.sql - not connected

SQLQuery18.sql - not connected

Displaying 18 Events

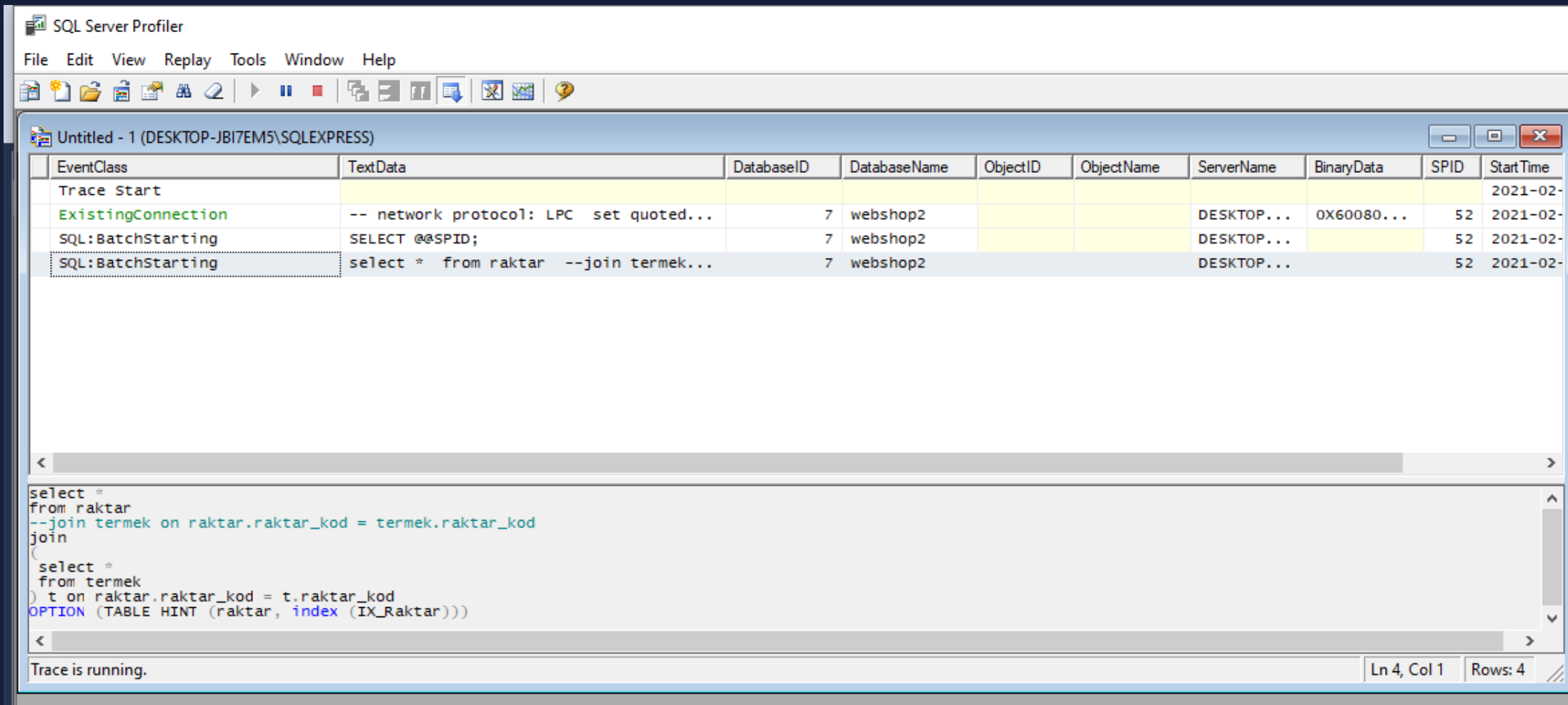
eve...	name	[TextData]	client app name	nt username	cpu time	logical re...	writes	duration	session id	timestamp
34	sql_batch_starting	USE [master]	Microsoft SQL Server ...	BKAE\gmolnar	NULL	NULL	NULL	NULL	75	2022-02-18 12:47:03....
35	sql_batch_completed	USE [master]	Microsoft SQL Server ...	BKAE\gmolnar	0	0	0	374	75	2022-02-18 12:47:03....
36	logout	NULL	SQLServerCEIP	NT SERVIC...	0	356	0	289783...	73	2022-02-18 12:47:41....
37	login	-- network protocol: LPC set quoted_identifi...	SQLServerCEIP	NT SERVIC...	NULL	NULL	NULL	NULL	62	2022-02-18 12:47:52....
38	sql_batch_starting	SET DEADLOCK_PRIORITY -10	SQLServerCEIP	NT SERVIC...	NULL	NULL	NULL	NULL	62	2022-02-18 12:47:52....
39	sql_batch_completed	SET DEADLOCK_PRIORITY -10	SQLServerCEIP	NT SERVIC...	0	0	0	94	62	2022-02-18 12:47:52....
40	sql_batch_starting	SELECT target_data FROM sys.dm_x...	SQLServerCEIP	NT SERVIC...	NULL	NULL	NULL	NULL	62	2022-02-18 12:47:52....
41	sql_batch_completed	SELECT target_data FROM sys.dm_x...	SQLServerCEIP	NT SERVIC...	187000	0	0	228279	62	2022-02-18 12:47:52....
42	logout	NULL	SQLServerCEIP	NT SERVIC...	187000	0	0	337000	62	2022-02-18 12:47:52....
43	rpc_completed	exec sp_reset_connection	SQLServerCEIP	NT SERVIC...	0	0	0	57	62	2022-02-18 12:47:52....
44	login	-- network protocol: LPC set quoted_identifi...	SQLServerCEIP	NT SERVIC...	NULL	NULL	NULL	NULL	62	2022-02-18 12:47:52....
45	sql_batch_starting	SET DEADLOCK_PRIORITY -10	SQLServerCEIP	NT SERVIC...	NULL	NULL	NULL	NULL	62	2022-02-18 12:47:52....
46	sql_batch_completed	SET DEADLOCK_PRIORITY -10	SQLServerCEIP	NT SERVIC...	0	0	0	100	62	2022-02-18 12:47:52....
47	sql_batch_starting	if not exists (select * from sys.dm_xe_sessio...	SQLServerCEIP	NT SERVIC...	NULL	NULL	NULL	NULL	62	2022-02-18 12:47:52....
48	sql_batch_completed	if not exists (select * from sys.dm_xe_sessio...	SQLServerCEIP	NT SERVIC...	16000	20	0	5012	62	2022-02-18 12:47:52....

Event:logout (2022-02-18 12:47:03.5470096)

Details

Field	Value
attach_activity_...	79B4153E-5ACB-4E82-B21A-0912E8F92C88
attach_activity_...	1
attach_activity_...	BC295C79-7132-4CBD-AC2B-AB26671D45D4
attach_activity_...	0
client_app_na...	Microsoft SQL Server Management Studio
client_pid	18988
cpu_time	1312000
duration	208336000
event_sequence	31
is_cached	True
is_dac	False
is_recovered	False
logical_reads	1993

SQL Server Profiler



SQL Server Profiler

File Edit View Replay Tools Window Help

Untitled - 1 (DESKTOP-JBI7EM5\SQLEXPRESS)

EventClass	TextData	DatabaseID	DatabaseName	ObjectID	ObjectName	ServerName	BinaryData	SPID	StartTime
Trace Start									2021-02-
ExistingConnection	-- network protocol: LPC set quoted...	7	webshop2			DESKTOP...	0X60080...	52	2021-02-
SQL:BatchStarting	SELECT @@SPID;	7	webshop2			DESKTOP...		52	2021-02-
SQL:BatchStarting	select * from raktar --join termék...	7	webshop2			DESKTOP...		52	2021-02-

```

select *
from raktar
--join termék on raktar.raktar_kod = termék.raktar_kod
join
(
  select *
  from termék
) t on raktar.raktar_kod = t.raktar_kod
OPTION (TABLE HINT (raktar, index (IX_Raktar)))
  
```

Trace is running.

Ln 4, Col 1 Rows: 4

Az Azure SQL ezt a szolgáltatást nem támogatja. Opcionálisan az Azure Data Studio Profiler Extension használható helyette