

Univerzitet u Novom Sadu,

Fakultet tehničkih nauka

SEMINARSKI RAD

Nastavni predmet: Administracija baza podataka

Naziv teme: Alat MS SQL Server Profiler

Predmetni nastavnik,

Bender mr Miroslav

Student,

Ana Domonji SR 46/2021

Maj, 2024

Sadržaj

1. Uvod u MS SQL Server Profiler	3
2. Konfiguracija i pokretanje Profilera	5
3. Analiza izvršavanja upita	11
4. Praćenje događaja sistema	13
5. Podešavanje filtera i sesija	14
7. Studije slučaja i primeri upotrebe	18
8. Zaključak	27

1. Uvod u MS SQL Server Profiler

MS SQL Server Profiler je alat za grafički korisnički interfejs (GUI) koja omogućava da pratimo i nadgledamo događaje koji se dešavaju u instanci SQL Servera. Može se koristiti za snimanje i čuvanje podataka o svakom događaju u datoteci ili tabeli radi kasnije analize. Na primer, možemo da nadgledamo proizvodno okruženje da bismo videli koje uskladištene procedure utiču na performanse tako što se izvršavaju presporo.

Predstavlja nezamenjivu komponentu za praćenje i analizu aktivnosti unutar SQL Server baza podataka. Izuzetno je važan u održavanju optimalne performanse i sigurnosti baza podataka.

MS SQL Server Profiler omogućava detaljno praćenje svih događaja koji se dešavaju unutar baze podataka, uključujući upite, izvršavanje procedura, povezivanje korisnika i mnoge druge aktivnosti. Kroz analizu ovih događaja, administratori mogu identifikovati potencijalne probleme, optimizovati performanse sistema i osigurati adekvatnu bezbednost podataka.

Glavni elementi MS SQL Server Profilera obuhvataju širok spektar funkcionalnosti, uključujući mogućnost filtriranja događaja, definisanja prilagođenih šablona za praćenje, generisanja izveštaja i još mnogo toga. Ova fleksibilnost omogućava prilagođavanje Profilera specifičnim potrebama svake baze podataka i organizacije.

SQL Server Profiler može se koristiti za aktivnosti kao što su:

- Analiza problematičnih upita radi pronalaženja uzroka problema.
- Pronalaženje i dijagnostikovanje sporo izvršavajućih upita.
- Zabeležavanje niza Transact-SQL naredbi koje dovode do problema.
- Rešavanje problema s performansama.
- Optimizacija upita.
- Revizija aktivnosti baze podataka.

SQL Server Profiler pruža niz ključnih funkcionalnosti koje omogućavaju praćenje i analizu događaja unutar SQL Servera.

Kroz ovaj alat, korisnici mogu zabeležiti širok spektar događaja koji se dešavaju u okviru instance SQL Servera. Ovi događaji mogu biti razvrstani u različite kategorije, uključujući T-SQL upite, uskladištene procedure, greške, upozorenja, zaključavanja i druge. Svaki zabeleženi događaj može biti obogaćen dodatnim informacijama putem podataka o kolonama. Ove informacije pružaju detaljan uvid u događaje, uključujući SQL tekst, trajanje, klijentsku aplikaciju, korisničko ime i slično.

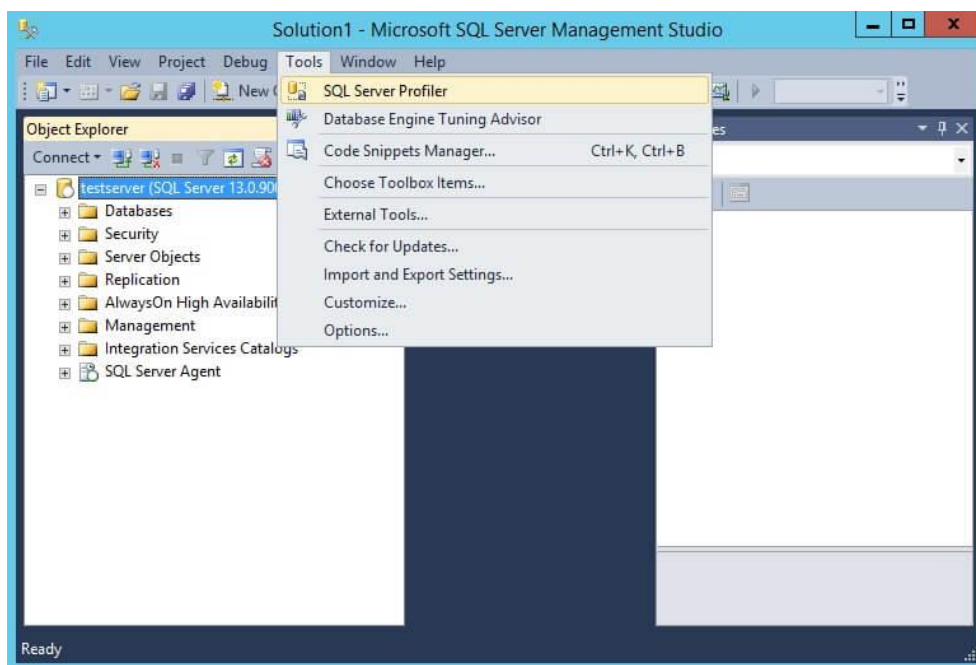
Filteri omogućavaju korisnicima da usmere svoju analizu na specifične aktivnosti ili transakcije. Mogu se primeniti različiti kriterijumi filtriranja, poput imena baze podataka, imena aplikacije, trajanja događaja i drugih.

Takođe pruža unapred definisane šablone za česte scenarije praćenja, olakšavajući postavljanje traga za različite svrhe, poput rešavanja problema s performansama ili revizije aktivnosti. Ima i mogućnost čuvanja i učitavanja tragova. Nakon konfigurisanja traga, korisnici mogu sačuvati zabeležene podatke za buduću analizu ili deljenje s drugima u cilju zajedničkog rešavanja problema. Omogućava i praćenje događaja u realnom vremenu, što omogućava korisnicima da odmah vide aktivnosti baze podataka dok se dešavaju. Ova funkcionalnost pruža trenutni uvid u stanje sistema i dodaje brzou reakciju na potencijalne probleme ili izazove.

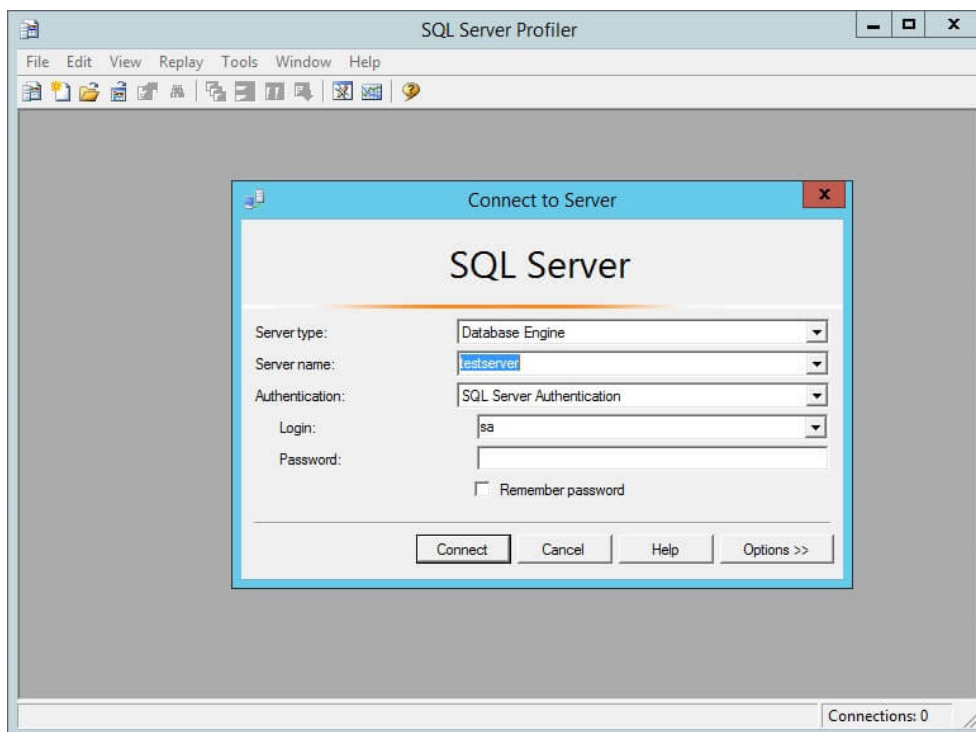
2. Konfiguracija i pokretanje Profilera

Konfiguracija i pokretanje MS SQL Server Profilera su ključni koraci u procesu efikasnog praćenja aktivnosti baze podataka. U nastavku su opisani osnovni koraci za konfigurisanje i pokretanje Profilera:

1. Otvaranje MS SQL Server Profilera: Prvo treba otvoriti SQL Server Management Studio (SSMS), aplikaciju koja se koristi za administraciju SQL Servera. Zatim, iz menija "Tools" izabrati opciju "SQL Server Profiler" za pokretanje Profilera.



2. Povezivanje sa SQL Server instancom: Nakon otvaranja Profilera, potrebno je uspostaviti vezu sa odgovarajućom SQL Server instancom. Kada se aplikacija pokrene, biće prikazan dijalog koji će omogućiti da se povežete na server koji želite da profilirate. To se takođe postiže i izborom opcije "File" -> "Connect" i zatim unošenjem informacija o serveru, kao što su ime servera, tip autentikacije i pristupni podaci. Kada popunite detalje o vezi, kliknite na „Poveži se“. Najbolja praksa je da se povezujemo sa udaljenim serverom, a ne da ovo pokrećemo na lokalnoj instanci (localhost) SQLServera.



3. Podešavanje opcija praćenja

Posle povezivanja sa serverom, prikazaće se prozor "Trace Properties" koji ima dve kartice. Prva kartica, „General“ koja omogućava:

- Imenovanje traga (trace).
- Biranje šablona za prikupljanje podataka. Šabloni su u osnovi sačuvani skup događaja i kolona podataka koje želite da pratite. Omogućavaju da se brzo pokrene praćenje bez potrebe za ručnim biranjem svih događaja i kolona podataka svaki put kada započne nova sesija praćenja. Microsoft nudi gomilu šablona i podrazumevano je izabran šablon „Standard“.
- Definisane gde da se sačuvaju podaci praćenja ako ne želimo samo da ih vidimo u GUI. Ima dve opcije za čuvanje. Prva opcija je da sačuvamo podatke u datoteku i navedemo parametre prelaska (skrivenne opcijama šablona). I drugu opciju da sačuvamo podatke u tabeli i navedemo broj redova u tabeli.
- Definisane vremena zaustavljanja praćenja.

Trace Properties

General | Events Selection

Trace name: Untitled - 1

Trace provider name: testserver

Trace provider type: Microsoft SQL Server "2016" CTP3.2 version: 13.0.900

Use the template: Standard (default)

☐ Save to file:

☐ Save to table:

☐ Set maximum rows (in thousands): 1

☐ Enable trace stop time: 2/27/2016 2:21:26 PM

Run Cancel Help

Druga kartica u ovom prozoru, „Events Selection“, je mesto gde možemo da konfigurišete koje događaje želimo da nadgledamo i koje kolone podataka želimo da uključimo za svaki događaj. Na ovoj kartici takođe možemo dodati filtere i promeniti način na koji su podaci organizovani u donjem desnom uglu interfejsa. U nastavku možemo videti da su, pošto smo izabrali šablon „Standard“ na prethodnoj kartici, kolone događaja i podataka već popunjene. U ovom trenutku ovo možemo urediti po potrebi ili jednostavno kliknite na „Run“ da biste započeli praćenje kakvo jeste.

Trace Properties

General | Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Events	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcess
Security Audit									
<input checked="" type="checkbox"/> Audit Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Audit Logout		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sessions									
<input checked="" type="checkbox"/> ExistingConnection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input type="checkbox"/>	<input checked="" type="checkbox"/>
Stored Procedures									
<input checked="" type="checkbox"/> RPC Completed	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TSQL									
<input checked="" type="checkbox"/> SQL Batch Completed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SQL Batch Starting	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>

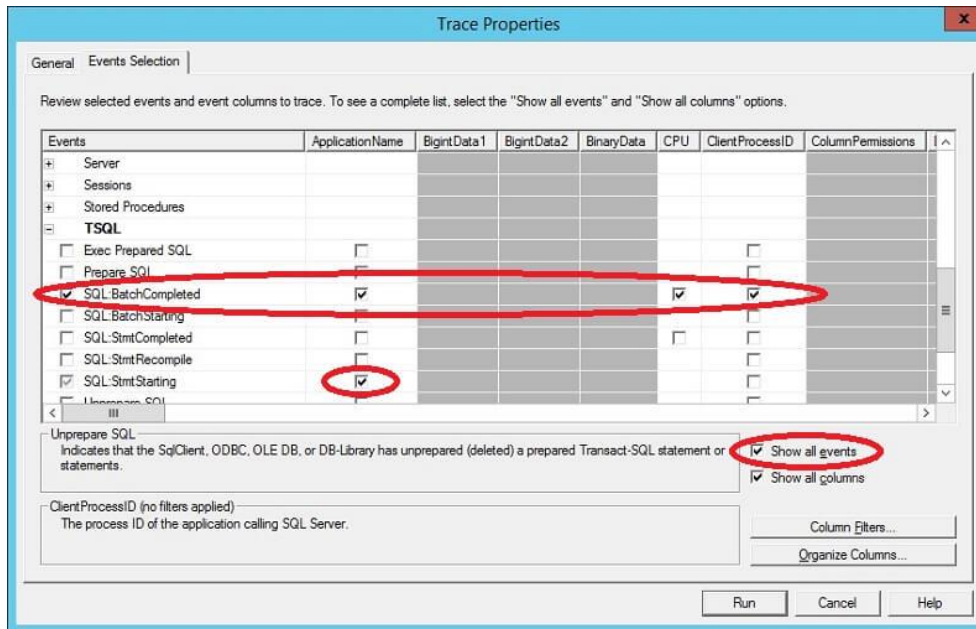
Security Audit
Includes event classes that are used to audit server activity.

☐ Show all events
☐ Show all columns

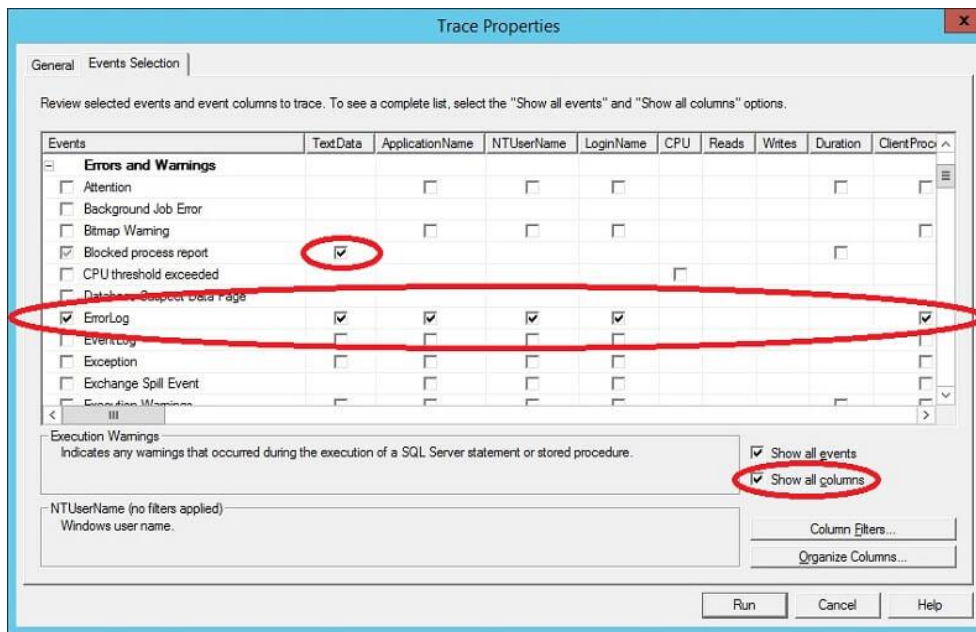
Column Filters...
Organize Columns...

Run Cancel Help

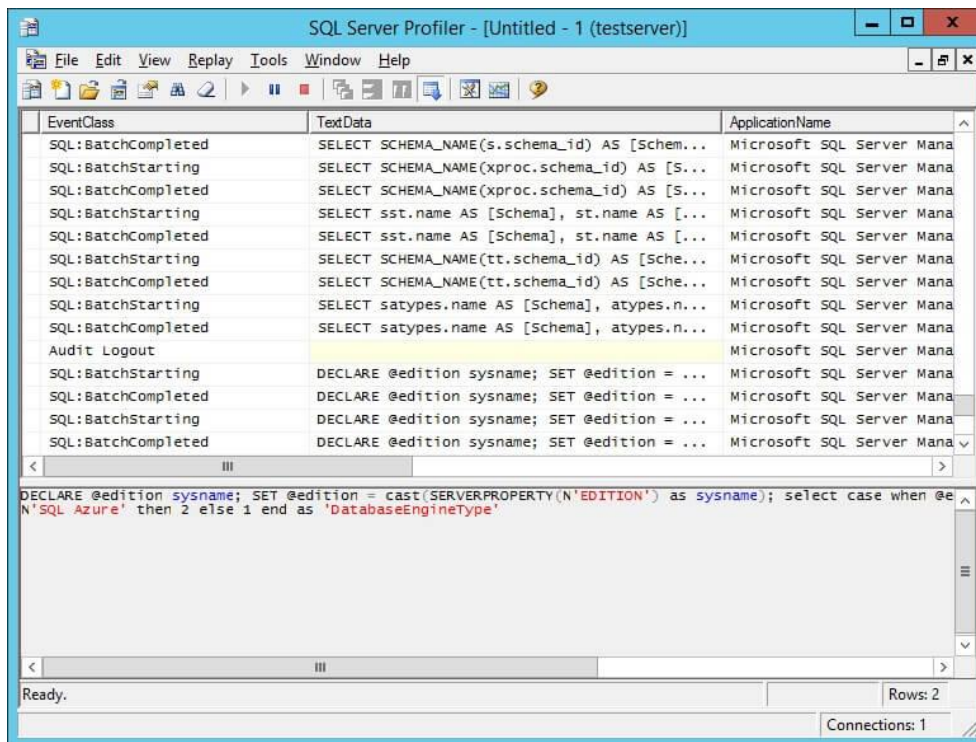
Da bismo dodali događaje sesiji praćenja, postoji polje za potvrdu „Show all events“ u donjem desnom uglu ekrana. Ako označimo ovo polje, prikazaće se svi događaji u prozoru iznad.



Takođe ima polje „Show all columns“ koje kada izaberemo događaj, sve kolone sa podacima budu izabrane. Kada se prikažu svi događaji, jednostavno proverimo događaje koje želimo da uključimo u naš trag. Da bismo izabrali kolone podataka, možemo ili da izaberemo samu kolonu podataka ili da izaberemo događaj koji će proveriti svaku kolonu podataka za taj događaj.



4. Pokretanje praćenja: Nakon konfigurisanja Profilera, možemo pokrenuti praćenje klikom na dugme "Run" ili pritiskom na odgovarajuću tastaturu. Profiler će početi da beleži sve definisane događaje koji se odvijaju unutar SQL Server instance. Nakon klika na dugme, vidi se ekran ispod i kako se događaji snimaju oni će biti prikazani u ovom prozoru.



Uz ove osnovne korake za konfiguraciju i pokretanje MS SQL Server Profilera, administratori mogu efikasno pratiti i analizirati aktivnosti baze podataka u cilju održavanja optimalnih performansi i sigurnosti.

3. Analiza izvršavanja upita

MS SQL Server Profiler omogućava detaljno praćenje izvršavanja SQL upita na serveru, pružajući administratorima važne informacije o performansama sistema i identifikaciju potencijalnih problema. Evo kako se to može demonstrirati:

Prvo je potrebno pokrenuti MS SQL Server Profiler i povezati se sa odgovarajućom instancom SQL Servera. Zatim, možemo definisati filtere kako bismo pratili samo SQL upite koji nas zanimaju, na primer, možemo postaviti filter da prikaže samo upite iz određene baze podataka ili samo upite koji traju duže od određenog vremena.

Kada je Profiler konfigurisan sa odgovarajućim filterima, možemo pokrenuti praćenje klikom na dugme "Run". Profiler će sada početi da beleži sve SQL upite koji se izvršavaju na serveru u realnom vremenu.

Interpretacija rezultata praćenja: Kada završimo praćenje, možemo analizirati rezultate kako bismo dobili uvid u izvršavanje SQL upita. Ključne informacije koje možemo pronaći u rezultatima uključuju:

- Vremensko trajanje izvršavanja: Ovo je vreme koje je svaki SQL upit proveo izvršavajući se na serveru. Duže trajanje može ukazivati na potencijalne probleme sa performansama.
- Broj izvršavanja: Koliko puta je određeni SQL upit izvršen u toku praćenja. Visok broj izvršavanja može ukazivati na upite koji se često koriste ili na potrebu za optimizacijom upita.
- Resursi korišćeni za svaki upit: Profiler može prikazati informacije o korišćenju resursa kao što su CPU, memorija i disk za svaki SQL upit. Ovo omogućava identifikaciju upita koji zahtevaju velike resurse i potencijalno opterećuju server.

U jednom praćenju(trace) možemo videti informacije kao što su:

- EventClass: Ovo je numerička vrednost koja označava tip događaja koji se dešava, kao što su SQL upiti, stored procedure, greške, i drugi.
- TextData: Ovo polje sadrži tekstualne podatke povezane sa svakim događajem, na primer SQL upit ili naziv stored procedure.
- ApplicationName: Ovo polje sadrži naziv aplikacije koja je izvršila SQL upit.
- NTUserName: Korisničko ime operativnog sistema koje je izvršilo SQL upit.
- Login Name: Korisničko ime koje se koristi za prijavu na SQL Server.
- CPU: Ovo je brojana vrednost koja predstavlja količinu CPU resursa korišćenih za izvršavanje upita.
- Reads: Ovo polje sadrži brojčane vrednosti koje predstavljaju broj čitanja sa diskova tokom izvršavanja upita.

- Writes: Ovo polje sadrži brojčane vrednosti koje predstavljaju broj zapisa na disk tokom izvršavanja upita.
- Duration: Ovo polje sadrži vremenski period izvršavanja upita, izražen u milisekundama.
- ClientProcessID: Ovo je identifikator procesa klijenta koji je izvršio SQL upit.
- SPID (Server Process ID): Ovo je identifikator procesa na SQL Serveru koji je izvršio SQL upit.
- StartTime: Ovo polje označava početak izvršavanja SQL upita.
- EndTime: Ovo polje označava kraj izvršavanja SQL upita.
- BinaryData: Ovo polje može sadržavati binarne podatke vezane za određene događaje.

Trace (LAPTOP-SC9RDVNE)

ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime	EndTime	BinaryData
SQLServerCEIP	SQLTELE...	NT SER...					12540	68	2024-05-15 14:55:52...		
SQLServerCEIP	SQLTELE...	NT SER...					12540	68	2024-05-15 14:55:52...		
SQLServerCEIP	SQLTELE...	NT SER...	0	0	0	0	12540	68	2024-05-15 14:55:52...	2024-05-15 14:55:52...	
SQLServerCEIP	SQLTELE...	NT SER...					12540	68	2024-05-15 14:55:52...		
SQLServerCEIP	SQLTELE...	NT SER...	31	631	0	188	12540	68	2024-05-15 14:55:52...	2024-05-15 14:55:52...	
SQLServerCEIP	SQLTELE...	NT SER...	31	2019	0	237	12540	68	2024-05-15 14:55:52...	2024-05-15 14:55:52...	
SQLServerCEIP	SQLTELE...	NT SER...	0	0	0	0	12540	68	2024-05-15 14:55:52...	2024-05-15 14:55:52...	0x00000...
SQLServerCEIP	SQLTELE...	NT SER...					12540	68	2024-05-15 14:55:52...		
SQLServerCEIP	SQLTELE...	NT SER...					12540	68	2024-05-15 14:55:52...		
SQLServerCEIP	SQLTELE...	NT SER...	0	0	0	0	12540	68	2024-05-15 14:55:52...	2024-05-15 14:55:52...	
SQLServerCEIP	SQLTELE...	NT SER...					12540	68	2024-05-15 14:55:52...		
SQLServerCEIP	SQLTELE...	NT SER...	0	20	0	4	12540	68	2024-05-15 14:55:52...	2024-05-15 14:55:52...	

if not exists (select * from sys.dm_xe_sessions where name = 'telemetry_xe_events')
alter event session telemetry_xe_events on server state=start

Trace is running.

Ln 127, Col 1 Rows: 127

Kombinovanjem ovih informacija, administratori mogu identifikovati upite koji mogu biti problematični za performanse sistema i preduzeti odgovarajuće korake za optimizaciju, kao što su indeksiranje tabela, restrukturiranje upita ili podešavanje konfiguracija servera.

4. Praćenje događaja sistema

MS SQL Server Profiler omogućava detaljno praćenje različitih događaja sistema unutar SQL Servera, što je od izuzetne važnosti za dijagnostiku, analizu performansi i otkrivanje problema. Evo kako ovi ključni događaji mogu biti korisni:

1. Povezivanje korisnika: Praćenje događaja povezivanja korisnika omogućava administratorima da prate ko se povezuje sa SQL Serverom, kada i sa kojim pristupnim pravima. Ovo je korisno za nadgledanje aktivnosti korisnika, identifikaciju potencijalnih problema sa autorizacijom i obezbeđivanje sigurnosti sistema.
2. Promene u strukturi baze podataka: Kada dođe do promena u strukturi baze podataka, kao što su kreiranje, modifikacija ili brisanje tabela, indeksa ili procedura, Profiler može zabeležiti ove događaje. To omogućava administratorima da prate sve strukturne promene i reaguju na njih na odgovarajući način, kao što su osiguravanje integriteta podataka ili analiza uticaja promena na performanse sistema.
3. Greške i upozorenja: Praćenje grešaka i upozorenja omogućava administratorima da brzo reaguju na potencijalne probleme sa radom SQL Servera. Profiler može zabeležiti različite vrste grešaka, kao što su greške pristupa podacima, problemi sa memorijom ili upozorenja o preopterećenju servera. Ovi događaji su ključni za dijagnostiku problema i sprečavanje daljih komplikacija.
4. Performanse i resursi: Pored specifičnih događaja, Profiler omogućava praćenje performansi sistema i korišćenja resursa kao što su CPU, memorija, disk i mreža. Ovi podaci su od vitalnog značaja za analizu performansi, identifikaciju uskih grla i optimizaciju konfiguracija servera radi poboljšanja performansi.

Kombinovanjem praćenja ovih ključnih događaja sistema, administratori mogu efikasno dijagnostikovati probleme, analizirati performanse sistema i preduzeti odgovarajuće korake za održavanje stabilnosti, sigurnosti i optimalnih performansi SQL Servera.

5. Podešavanje filtera i sesija

Podešavanje filtera i sesija u MS SQL Server Profileru omogućava fokusiranje na specifične događaje ili aktivnosti kako bi se efikasno pratili samo relevantni podaci. Evo kako se koriste filteri i sesije:

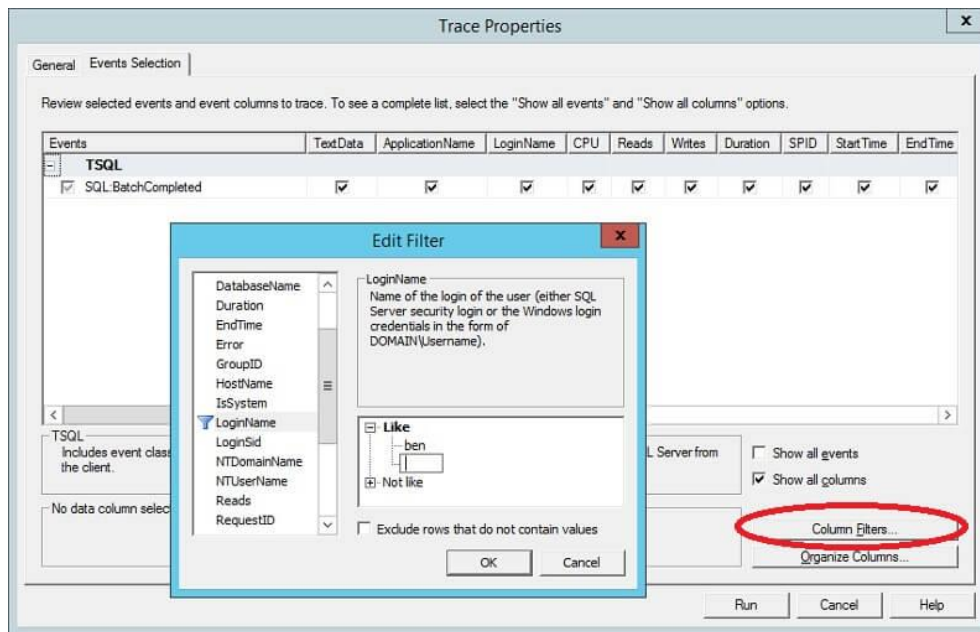
Filteri:

Filteri omogućavaju da filtriramo događaje koji su uhvaćeni u našoj sesiji SQL praćenja. Koriste se za ograničavanje prikaza podataka na samo one događaje ili aktivnosti koje su od interesa.

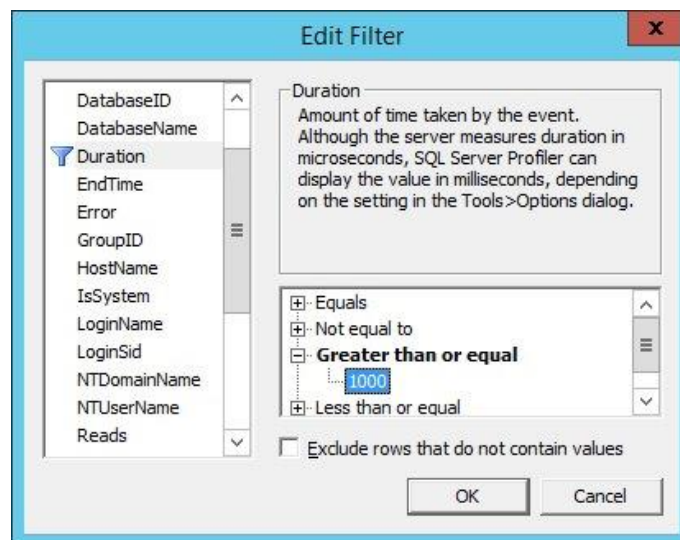
Postoji nekoliko vrsta filtera koji se mogu primeniti u SQL Server Profileru:

1. Event Filters (Filteri događaja): Ovi filteri omogućavaju da odaberemo tačno koji događaji će biti prikazan u tragu. Na primer, možemo filtrirati samo SELECT upite ili samo greške.
2. Data Filters (Filteri podataka): Ovi filteri omogućavaju da postavimo uslove na osnovu vrednosti određenih podataka koji su uhvaćeni u tragu. Na primer, možemo filtrirati samo transakcije koje se odnose na određenog korisnika ili samo transakcije koje imaju određeni vremenski okvir.
3. Column Filters (Filteri kolona): Ovi filteri omogućavaju da postavimo uslove na osnovu vrednosti određenih kolona podataka koje su uhvaćene u tragu. Na primer, možemo filtrirati samo događaje koji sadrže određeni tekst ili samo događaje koji imaju određenu vrednost u određenoj koloni.

Filter se može primeniti na bilo koju kolonu podataka koja je deo događaja koji smo izabrali. Čak i ako kolona podataka nije sačuvana kao deo sesije SQL praćenja, i dalje možemo primeniti filter na osnovu nje. Da bismo napravili filter, biramo dugme „Column Filters“ na kartici „Events Selection“. Kada izaberemo "Edit Filter" biće prikazan dijalog. U slučaju ispod izabrana je kolonu podataka LoginName koja pruža klauzule za poređenje teksta i filtrirano je LoginName „ben“. Jedna stvar koju treba imati na umu kada koristite operator LIKE je da ako ne dodamo „%“, on se tretira kao operator jednakosti (=). Na primer, ako želimo da tražimo bilo koju naredbu koju izvršava prijava koja je počela sa „app“, imaćemo „app%“ kao filter.



U zavisnosti od kolone podataka koju izaberemo, videćemo različite operatore poređenja. Kao što vidimo u ovom primeru ispod, kolona podataka o trajanju sadrži numeričke klauzule za poređenje.



Pravilno primenjeni filteri omogućavaju vam da se fokusirate samo na relevantne informacije, čime se olakšava analiza traga i pronalaženje potencijalnih problema.

Sesije:

Sesije u SQL Server Profileru predstavljaju konfiguraciju traga koja definiše koje događaje ili aktivnosti treba pratiti i kako treba prikazati rezultate. Kada konfigurišemo sesiju, možemo

odabrati koje događaje želimo pratiti, koje kolone podataka želimo da prikazemo, kao i da li želimo da primenite filtere na tragu.

Svaka sesija može imati svoje specifične postavke i filtere, što omogućava da se prilagode različitim scenarijima praćenja. Na primer, možemo imati sesiju koja prati samo upite koji traju duže od određenog vremenskog perioda, dok druga sesija može pratiti samo specifične vrste grešaka.

Kombinovanjem filtera i sesija, administratori mogu efikasno pratiti i analizirati aktivnosti u bazi podataka, fokusirajući se samo na relevantne informacije koje su im potrebne za dijagnostiku problema ili optimizaciju performansi.

6. Analiza rezultata i dijagnostika problema

Analiza rezultata dobijenih korišćenjem MS SQL Server Profilera ključna je za identifikaciju potencijalnih problema sa performansama, bezbednošću ili efikasnosti upita. Evo nekoliko koraka za analizu rezultata i dijagnostiku problema:

1. Identifikacija sporih upita: Pregledajte rezultate traga kako biste identifikovali upite koji uzrokuju zastoje ili usporeno izvršavanje. Obratite pažnju na upite koji imaju dugu izvršnu vreme ili visok broj izvršavanja. To može ukazivati na potrebu za indeksiranjem određenih tabela ili optimizacijom upita.
2. Analiza bezbednosnih problema: Proverite da li su u tragu zabeležene neke sumnjive aktivnosti koje bi mogle ukazivati na pokušaj neovlašćenog pristupa ili zloupotrebu sistema. Obratite pažnju na neuspele pokušaje prijavljivanja, neobične SQL izjave ili bilo kakve druge aktivnosti koje nisu tipične za vašu aplikaciju.
3. Procena efikasnosti upita: Uporedite rezultate traga sa očekivanim performansama vaših upita. Ako primetite značajne razlike u izvršno vreme ili broju izvršavanja, to može ukazivati na potrebu za optimizacijom upita ili poboljšanjem dizajna baze podataka.
4. Dijagnostika problema: Koristite detaljne informacije prikupljene u tragu kako biste identifikovali uzrok problema. Obratite pažnju na SQL kod koji se izvršava, vreme izvršavanja, broj izvršavanja, kao i eventualne greške ili upozorenja. Ovo će vam pomoći da bolje razumete šta uzrokuje probleme i kako ih rešiti.
5. Optimizacija performansi: Na osnovu analize rezultata traga, možete doneti odluke o tome kako optimizovati performanse vašeg sistema. To može uključivati indeksiranje tabela, optimizaciju SQL upita, podešavanje konfiguracija servera ili čak optimizaciju samog aplikacijskog koda.

Korišćenjem MS SQL Server Profilera za analizu rezultata, možete brzo identifikovati probleme i preduzeti odgovarajuće korake kako biste poboljšali performanse, bezbednost i efikasnost vaših SQL Server baza podataka.

7. Studije slučaja i primeri upotrebe

Postoji opšte uverenje da bilo koja aplikacija, kao i bilo šta u stvarnom svetu, nije idealna i da uvek postoji nešto što treba razviti, učiniti boljom i konkurentnijom. Konkretno, u razvojnim okruženjima, kod se obično češće menja kako bi se rešili problemi, ispunili poslovni ciljevi, povećali performanse itd. Stoga su za mnoge programere istraživanje problema, razumevanje problema i ispravljanje grešaka deo njihove svakodnevne rutine. Za aplikacije zasnovane na bazi podataka, ponekad je teško razumeti da li je problem na strani baze podataka ili na strani aplikacije. U takvim slučajevima, programeri baze podataka i aplikacija treba da sarađuju kako bi lako analizirali, locirali i rešili problem.

Pretpostavlja se da naša aplikacija pristupa bazi podataka samo pozivanjem uskladištenih procedura. Dakle, da bismo pronašli problem, moramo analizirati pozive odgovarajućih uskladištenih procedura. U nastavku će biti opisano kako se koristiti SQL Server Profiler da bi se lako uhvatio određeni poziv uskladištene procedure i izbegli hvatanje nepotrebnih informacija u profiler. Posedovanje samo neophodnih informacija o pozivu procedure, učiniće proces otklanjanja grešaka jednostavnijim i bržim. Pa, ilustrovaćemo ovaj proces studijom slučaja. Počnimo sa kreiranjem testnog okruženja:

```
USE master
GO

CREATE DATABASE TestDB
GO

USE TestDB
GO

CREATE TABLE TestTable
(
    ID INT PRIMARY KEY,
    Val CHAR(1)
)
GO

INSERT INTO TestTable(ID, Val)
VALUES (1, 'A'), (2, 'B'), (3, 'C')
```

Napravili smo uzorak baze podataka sa uzorkom tabele. Sada ćemo kreirati nekoliko uskladištenih procedura koje poziva aplikacija za popunjavanje i preuzimanje podataka. Prvu poziva aplikacija da ubaci podatke u TestTable:

```

CREATE PROCEDURE uspInsertData
    @pID INT,
    @pVal CHAR(1),
    @ResCode VARCHAR(10) OUTPUT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION

        INSERT INTO TestTable(ID, Val)
        VALUES (@pID, @pVal)

        COMMIT

    END TRY
    BEGIN CATCH

        IF @@TRANCOUNT > 0
            ROLLBACK

    END CATCH

    SET @ResCode='OK'
END
GO

```

Druga procedura dobija red iz tabele po njegovoj vrednosti:

```

USE TestDB
GO

CREATE PROCEDURE uspGetData
    @pVal CHAR(1)
AS
BEGIN
    SELECT ID, Val
    FROM TestTable
    WHERE Val=@pVal
END

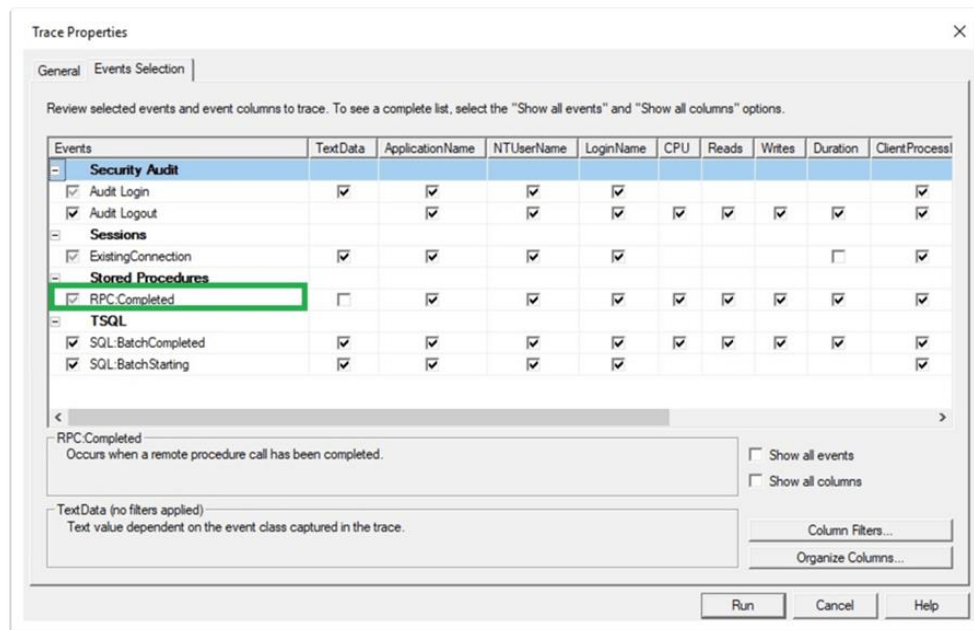
```

Sada, pretpostavimo da je jedan od programera aplikacije prijavio da pokušava da ubaci podatke u tabelu i dobije „OK“ iz baze podataka. Međutim, kada pretražuje podatke po ubačenoj vrednosti, ne dobija ništa. Pošto znamo da je procedura „uspInsertData“ pozvana da ubaci red u tabelu, pokrenućemo SQL Server Profiler da uhvati poziv ove procedure i zatraži od programera da reprodukuje problem. Da bismo to uradili, otvaramo SQL Server Management Studio (SSMS) i biramo „Alati“ > „SQL Server Profiler“:

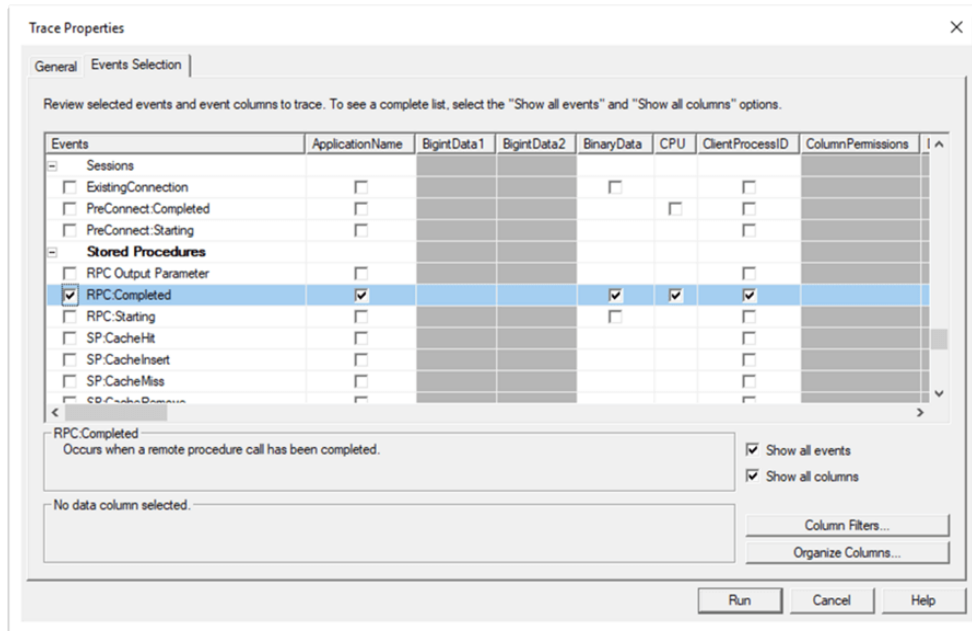
Kada se Profiler pokrene, povezujemo se sa našom instancom.

Ovde bi trebalo da izaberemo odgovarajući šablon za naš zadatak. Svaki šablon pruža grupu događaja koji imaju za cilj određenu svrhu. Kako treba da pratimo samo određeni poziv procedure i ne trebamo nikakve dodatne informacije, zainteresovani smo samo za prikupljanje

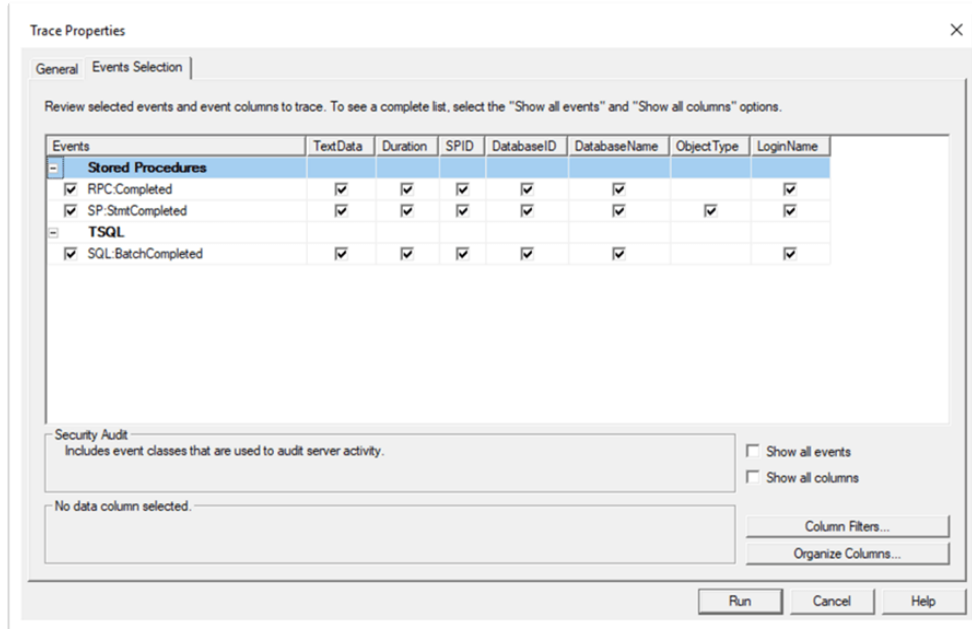
informacija o našem pozivu uskladištene procedure. Dakle, u ovom koraku se postavlja pitanje koji šablon treba da izaberemo. U stvari, možemo uhvatiti informacije o pozivu uskladištene procedure koristeći skoro svaki šablon. Međutim, pošto se naše procedure pozivaju sa strane aplikacije, mi se bavimo udaljenim pozivima procedura. Zbog toga smo zainteresovani za snimanje RPC (poziv udaljenog postupka) događaja. Na primer, možemo da izaberemo šablon „Standard“ (podrazumevano), a zatim poništimo neke nepotrebne događaje i ostavimo samo one koji se odnose na udaljene uskladištene procedure:



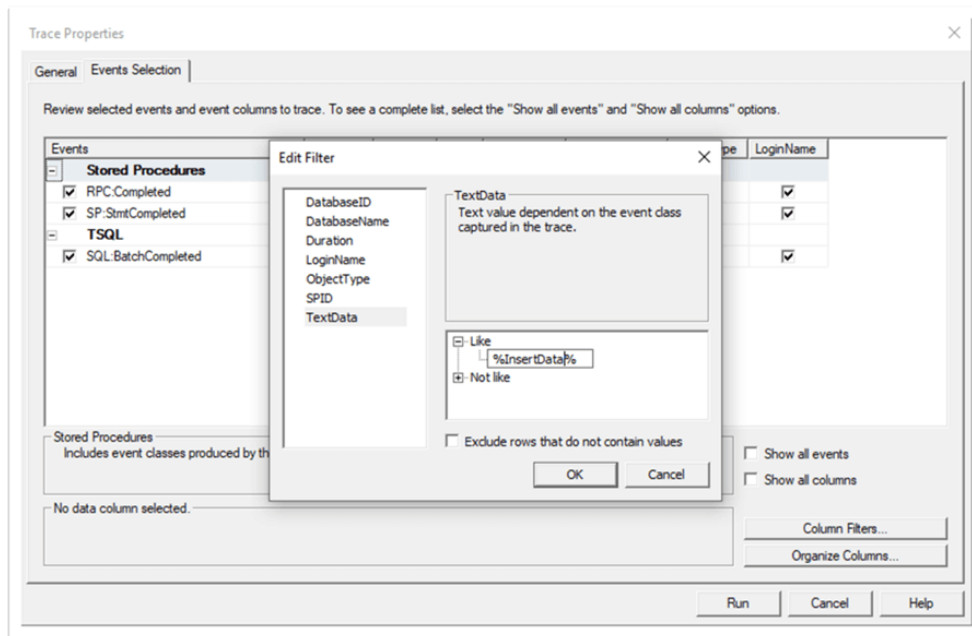
U našem konkretnom primeru, možemo ostaviti samo "RPC: Completed ". Za odabir samo događaja „RPC:Completed“ bez trošenja vremena na poništavanje izbora drugih, takođe možemo da koristimo prazan šablon i proverimo samo odgovarajući događaj:



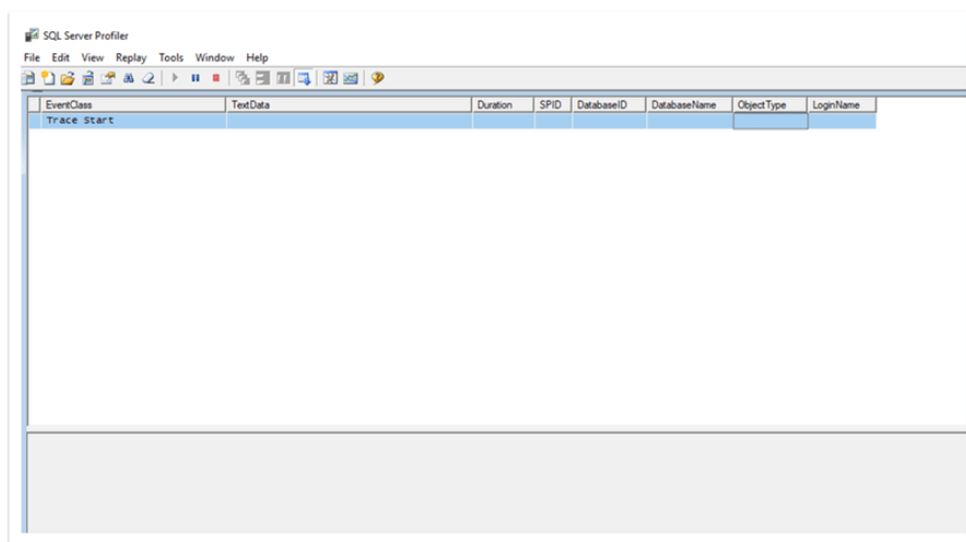
Dok istražujem pozive udaljenih uskladištenih procedura, biram šablon " Tuning" za analizu takve vrste zadatka jer postoji dosta događaja koji se uglavnom odnose na uskladištene procedure. Stoga, ne moramo ručno da poništavamo bilo šta kada biramo ovu opciju:



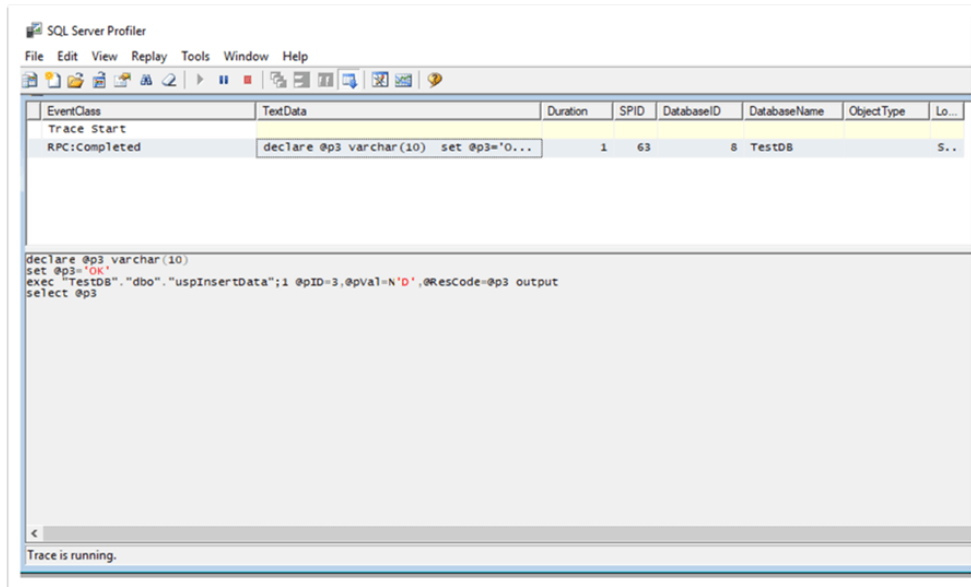
Da uhvatimo određenu sačuvanu proceduru, biramo „Column Filters...“ i postavljamo filter „TextData“. Ako ne znamo puno ime procedure, možemo da otkucamo samo njen deo:



Nakon podešavanja filtera, kliknemo na „Run“ da započnemo praćenje:



Nakon pokretanja praćenja, tražimo od programera da reprodukuje problem. Kada izvrši akciju koja poziva našu uskladištenu proceduru, možemo pronaći poziv na profileru. U našem primeru možemo pronaći izvršni kod naše procedure u „RPC: Completed“:

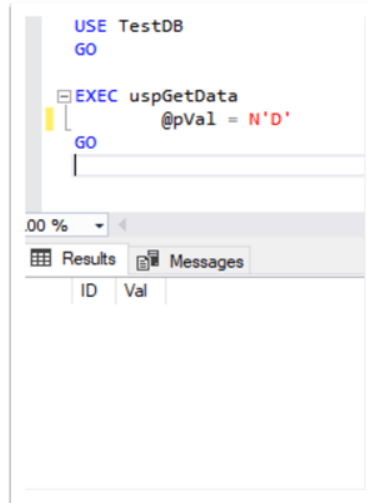


Tako smo uhvatili neophodan poziv sačuvane procedure. Sad pauziramo praćenje, kopiramo izvršni kod i nalepimo ga na SSMS da analiziramo šta se dešava:

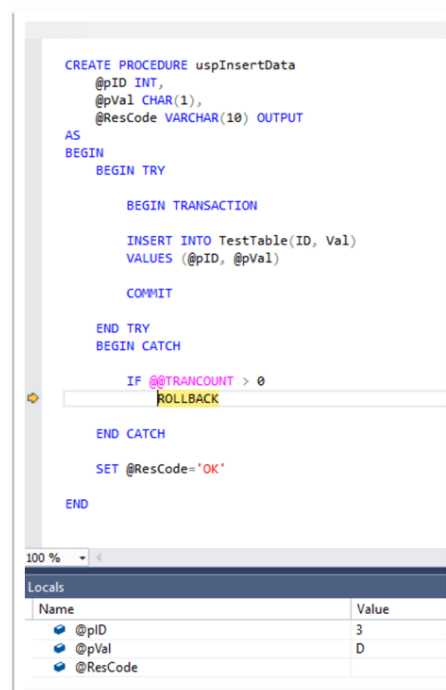
Ako izvršimo gornji kod u SSMS-u, možemo videti da procedura vraća „OK“:



Ipak, ako pokušamo da izaberemo redove sa Val=D, nećemo dobiti ništa:



Pa, da bismo pronašli problem, počinjemo da otklanjamo greške u snimljenom pozivu uskladištene procedure:



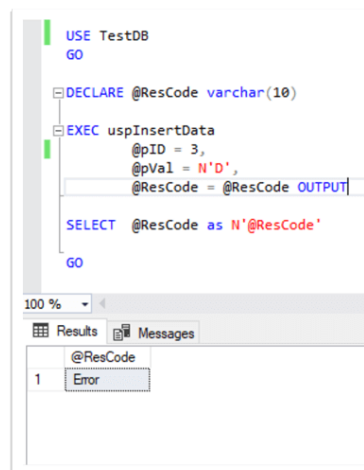
Vidimo da je procedura vraćena. To je zato što pokušavamo da ubacimo duplikat ID u tabelu, koji je jedinstven. Pored toga, jasno je iz otklanjanja grešaka da je procedura osmišljena na pogrešan način i uvek vraća „OK“ (čak i kada ne uspe). Dakle, problem je pronađen i možemo ga rešiti izmenom uskladištene procedure:


```

ALTER PROCEDURE uspInsertData
@pID INT,
@pVal CHAR(1),
@ResCode VARCHAR(10) OUTPUT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
        INSERT INTO TestTable(ID, Val)
        VALUES (@pID, @pVal)
        COMMIT
        SET @ResCode='OK'
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK
        SET @ResCode='Error'
    END CATCH
END
GO

```

Sada će programeri dobiti tačnu poruku o rezultatu procedure:



Ovo je veoma jednostavan primer, ali korak po korak ilustruje kako se SQL Server Profiler može koristiti za snimanje poziva uskladištene procedure koji je često neophodan za rešavanje problema vezanih za bazu podataka. Uprkos činjenici da su u stvarnim situacijama slučajevi obično mnogo komplikovaniji, pozive procedura možemo uhvatiti na isti jednostavan način, a zatim ih analizirati pomoću debagera.

U zaključku, SQL Server Profiler je moćan alat za praćenje poziva uskladištenih procedura. Poziv uhvaćene procedure može biti od pomoći u pronalaženju problema. Većina unapred definisanih šablona Profiler-a je obezbeđena sa događajima koji se odnose na uskladištene procedure. Konkretno, predefinisani šablon „Tuning” u SQL Server Profiler-u može biti veoma koristan ako treba da uhvatimo samo pozive udaljenih sačuvanih procedura. Dodatno, filter kolone „TextData“ se može koristiti za pronalaženje uskladištene procedure po njenom imenu (ili po delu njenog imena) i praćenje samo poziva te procedure. Dakle, moći ćemo da pratimo

samo poziv određene uskladištene procedure i izbegnemo generisanje dodatnih, nepotrebnih informacija.

8. Zaključak

U zaključku, MS SQL Server Profiler predstavlja temeljni alat za svakog administratora baza podataka koji želi da održi stabilnost, performanse i bezbednost svog SQL Server okruženja. Ovaj alat pruža dubok uvid u sve aktivnosti unutar baze podataka, omogućavajući precizno praćenje, analizu i dijagnostiku različitih aspekata sistema.

Kroz praćenje događaja kao što su SQL upiti, greške, bezbednosne aktivnosti i promene u strukturi, SQL Server Profiler omogućava administratorima da identifikuju i reše potencijalne probleme koji bi mogli da utiču na performanse ili bezbednost sistema. Njegova sposobnost da identifikuje sporije upite, dijagnostikuje greške u aplikacijama i pratiti promene u strukturi baze podataka omogućava efikasnu reakciju na probleme i poboljšanje performansi.

Kroz analizu rezultata traga, administratori mogu doneti informisane odluke o optimizaciji performansi, indeksiranju tabela, kao i implementaciji sigurnosnih mera radi zaštite sistema od neovlašćenih aktivnosti. MS SQL Server Profiler pruža neophodne alate i informacije za efikasno vođenje i održavanje SQL Server baza podataka, čineći ga ključnim resursom u arsenalu svakog administratora baza podataka. Svojom sposobnošću da pruži detaljne uvide i informacije, SQL Server Profiler postaje neprocenjiv alat za upravljanje i optimizaciju SQL Server sistema u svakom okruženju.