

Lab 1: Praćenje vrijednosti performance countera

Priprema za lab

Za ovaj lab pokrenite virtualni stroj **ABP-VM2** primjenom snapshota **Početo stanje**. Time ćete automatski biti prijavljeni na sustav kao **Student** s lozinkom **Pa\$\$w0rd**.

Cilj laba

U ovom labu naučit ćete koristiti alat **Performance Monitor** da biste snimili vrijednosti *performance countera* i kasnije ih analizirali. Vidjet ćete preko kojih se osnovnih *countera* može zaključiti kakvo je stanje procesora, memorije i diska, koliko ima aktivnosti na SQL Serveru te koliki dio ukupnog opterećenja sustava uzrokuje SQL Server.



Zadaci

Zadaci od kojih se ovaj lab sastoji su:

- Kreiranje *Data Collector Seta* i odabir *performance countera*
- Generiranje aktivnosti na serveru i snimanje *performance countera*
- Pregled snimljenih *performance countera*

Zadatak	Detaljni koraci																		
1. Kreiranje <i>Data Collector Seta</i> i odabir <i>performance countera</i>	<ul style="list-style-type: none">• Pokrenite Performance Monitor:<ul style="list-style-type: none">○ Kliknite na Start → Administrative Tools → Performance Monitor• Definirajte novi Data Collector Set:<ul style="list-style-type: none">○ U konzoli Performance Monitor, pod Data Collector Sets, desno kliknite User Defined i odaberite New Data Collector Set.○ Novi Data Collector Set nazovite DS1. Umjesto predloženog korištenja predložka, odaberite opciju Create manually (advanced).○ U sljedećem koraku odaberite Create data logs, Performance counter○ Dodajte sljedeće countere za snimanje: <table border="1"><thead><tr><th>Objekt</th><th>Counter</th><th>Instanca</th></tr></thead><tbody><tr><td>Processor</td><td>% Processor Time</td><td>_Total</td></tr><tr><td>Processor</td><td>% Privileged Time</td><td>_Total</td></tr><tr><td>Processor</td><td>% User Time</td><td>_Total</td></tr><tr><td>Memory</td><td>Available MBytes</td><td></td></tr><tr><td>Memory</td><td>Pages/sec</td><td></td></tr></tbody></table>	Objekt	Counter	Instanca	Processor	% Processor Time	_Total	Processor	% Privileged Time	_Total	Processor	% User Time	_Total	Memory	Available MBytes		Memory	Pages/sec	
Objekt	Counter	Instanca																	
Processor	% Processor Time	_Total																	
Processor	% Privileged Time	_Total																	
Processor	% User Time	_Total																	
Memory	Available MBytes																		
Memory	Pages/sec																		

	Physical Disc	% Disk Time	_Total
	Physical Disc	Avg. Disk Queue Length	_Total
	Physical Disc	Avg. Disk Sec/Read	_Total
	Physical Disc	Avg. Disk Sec/Write	_Total
	Process	% User Time	Total sqlservr, sqlservr#1, sqlservr#2
	Process	Working Set	Total sqlservr, sqlservr#1, sqlservr#2
	Process	IO Data Bytes/sec	Total sqlservr, sqlservr#1, sqlservr#2
	SQLServer:Buffer Manager	Page Life Expectancy	
	SQLServer:Buffer Manager	Buffer cache hit ratio	
	SQLServer:Buffer Manager	Page reads/sec	
	SQLServer:Buffer Manager	Page writes/sec	
	SQLServer:Memory Manager	Total Server Memory (KB)	
	SQLServer:Memory Manager	Target Server Memory (KB)	
	SQLServer:SQL Statistics	Batch Requests/sec	
	SQLServer:General Statistics	User Connections	
	<ul style="list-style-type: none"> ○ Pod Sample interval zadajte da se vrijednosti brojača snimaju svake 3 sekunde ○ Za Root directory prihvatite ponuđenu putanju ○ Prihvatite ponuđene opcije Run as: <Default> i Save and close te kliknite Finish ○ U mapi Data Collector Sets, pod User Defined, stvorena je nova stavka – DS1 		
2. Generiranje	<ul style="list-style-type: none"> ● Pokrenite log DS1: 		

<p>aktivnosti na serveru i snimanje <i>performance counter</i>a</p>	<ul style="list-style-type: none"> ○ Data Collector Sets → User Defined → DS1 → desni klik → Start ● Na taskbaru nalazi se ikona programa Workloader. Pokrenite ga. <p>Aplikacija Workloader spaja se na bazu AdventureWorks2014 i nad njom izvršava nekoliko SQL upita paralelno. Iz padajućih izbornika mogu se odabrati različiti scenariji. U svakom scenariju nalaze se drugačiji upiti, a traju otprilike 90 sekundi.</p> <ul style="list-style-type: none"> ● Odaberite sljedeći scenarij: <ul style="list-style-type: none"> ○ Upiti za blokiranje: 1 ○ Upiti za veliko opterećenje: 2 ● Kliknite gumb Pokreni upite ● Kad aplikacija Workloader završi s izvođenjem upita, u Performance Monitoru desno kliknite na DS1 i odaberite Stop.
<p>3. Pregled snimljenih <i>performance counter</i>a</p>	<ul style="list-style-type: none"> ● U Performance Monitoru otvorite izvještaj nastao ovim snimanjem: <ul style="list-style-type: none"> ○ Reports → User defined → DS1 → ABP-VM2_..... ● Početni prikaz vrijednosti countera je grafički. Promijenite ga tako da kliknete na strelicu na gumbu  i odaberete opciju Report. ● Performance Monitor prikazao je većinu countera koje smo mjerili, ali je neke ipak propustio – radi se o counterima iz kategorije Process koji se odnose na instance SQL Servera. Njih dodajte sami: ● Na alatnoj traci kliknite na gumb  i dodajte sve countera iz kategorije Process ● Sada imamo prikazanu prosječnu vrijednost svih countera koje smo mjerili i možemo ih analizirati. <p>Analiza vrijednosti countera</p> <p>Procesor</p> <p>Prosječna vrijednost countera Processor: % Processor Time je oko 57%. Možemo se držati grube procjene da procesor nije preopterećen dok je ta vrijednost manja od 80%.</p> <p>Pored countera %Processor Time dobro je promotriti i countera % Privileged Time i % User Time. Inače vrijedi: % Processor Time = % Privileged Time + % User Time. Drugim riječima, procesorsko vrijeme možemo razdvojiti na privileged i user vrijeme. Ako procesor radi u privileged (kernel) modu, to znači da se vjerojatno izvodi neki driver, I/O request packet ili da se događa context switching. Vrijednost veća od 30 ne bi bila dobra jer bi to značilo da se velik dio vremena troši na “sistemske” stvari pa da vjerojatno nema dovoljno vremena za izvođenje aplikacijskih procesa. U našem slučaju izmjereno je % Privileged Time = 4%, što je zadovoljavajuće.</p>

Memorija

Iz countera **Available MBytes** vidimo da na serveru imamo 909 MB slobodne memorije, što je dobro. No, counter **Pages/sec** ima vrijednost 47, što je dosta visoko. Inače, taj nam counter pokazuje koliko ima straničenja i jedan je od najboljih indikatora stanja memorije. Ako je njegova vrijednost konstantno iznad 20, to bi moglo ukazivati na problem, a vrijednosti preko 50 nisu dobre. No, ako je njegova vrijednost većinu vremena niska, a bilježi samo povremene skokove, to ne bi trebalo biti problematično. S obzirom da smo zabilježili visoku prosječnu vrijednost, trebat ćemo malo detaljnije analizirati kako se ona kretala kroz cijeli period mjerenja te pogledati i neke druge pokazatelje.

Disk

Counter **% Disk Time** trebao bi pokazivati koliko je posto vremena disk zauzet izvršavanjem ulazno-izlaznih operacija. Vrijednost 100% značila bi da je disk cijelo vrijeme zauzet. No, vrijednost tog countera ne dolazi iz izravnog mjerenja, nego se izračunava. Tako se ponekad može dogoditi da ta vrijednost bude i veća od 100%, što je bio slučaj i u ovom našem primjeru. Iz ovoga možemo pretpostaviti da je disk dosta opterećen, ali trebamo promotriti i neke druge countere. **Avg. Disk Queue Length** pokazuje koliko ima I/O zahtjeva u redu čekanja. Vrijednost koja je više nego dvaput veća od broja fizičkih diskova ukazuje na preopterećenost. Kod nas je izmjerena vrijednost 1.622, što bi moglo značiti da je disk dosta opterećen. **Avg.Disc sec/Read** i **Avg.Disc sec/Write** pokazuju koliko u prosjeku traje jedno čitanje, odnosno zapisivanje na disk. Vrijednost veća od 0,02 (20 milisekundi) ukazuje na problem. U našem primjeru vrijeme potrebno za zapisivanje nalazi se upravo na toj granici. Gledajući zajedno sve ove pokazatelje rada diska, čini se da je disk upravo na granici da bude preopterećen.

SQL Serverov udio u ukupnom opterećenju

Koliko određeni proces pridonosi ukupnom opterećenju servera može se vidjeti preko countera koji se nalaze u kategoriji **Process**.

Za svaku SQL Server instancu imamo poseban counter u svakoj od kategorija. Instance su označene sa sqlservr, sqlservr#1 i sqlservr#2 i ispostavlja se da je defaultna instanca ABP-VM2 označena s #2, iako to ne mora uvijek biti tako.

Da bismo vidjeli koliki je SQL Serverov udio u ukupnom opterećenju procesora, možemo pogledati counter **Process: % User Time**. Vidimo da od ukupnog user timea (_Total = 54,368) najveći dio otpada upravo na sqlservr#2 (51,878). Dakle, možemo zaključiti da opterećenje procesora gotovo u potpunosti dolazi od SQL Server procesa.

SQL Serverov udio u opterećenju memorije možemo iščitati iz countera **Process: Working Set**. Tu možemo primijetiti da od tri SQL Server instance najviše memorije

zauzima sqlservr#2 (300.000 od ukupno 1.000.000).

Diskovno opterećenje servera gotovo u potpunosti dolazi od procesa sqlservr#2. To se vidi iz countera **Process: IO Data Bytes/sec**

Stanje SQL Servera


Jedan od najvažnijih pokazatelja stanja SQL Servera je counter **Page life expectancy** iz kategorije **SQL Server:Buffer Manager**. On pokazuje koliko se procjenjuje da će se jedna stranica u prosjeku zadržati u međuspremniku. Kad bi sve stranice mogle biti cijelo vrijeme u međuspremniku, to bi značilo da je on dovoljno velik da cijela baza stane u memoriju. Budući da to obično nije slučaj, da bi SQL Server stranice s diska mogao staviti u međuspremnik, iz njega će morati istisnuti neke druge. Ako je vrijednost countera Page life expectancy veća od 300 (sekundi), možemo reći da je stanje dobro. U našem primjeru vrijednost je veća od 1000. Drugi vrlo važan pokazatelj stanja međuspremnika je **Buffer Cache Hit Ratio**. On pokazuje koliko je međuspremnik efikasan, tj. u kojem se postotku tražene stranice pronalaze već u međuspremniku pa ih ne treba tražiti na disku. Ova bi vrijednost trebala biti iznad 90 (%), a kod nas je 99,994, iz čega bismo mogli zaključiti da je situacija dobra.




Korisnu informaciju o stanju memorije mogu nam dati još dva countera iz kategorije SQL Server: Memory Manager – **Total server memory** i **Target server memory**. Total server memory pokazuje koliko memorije SQL Server trenutačno koristi, a Target server memory pokazuje koliko SQL Server misli da bi bilo optimalno. Trenutačno pod Target Server memory stoji vrijednost od oko 1 GB, koliko je postavljeno preko parametra Max server memory. No, izgleda da mu zasad uopće ne treba toliko je Total server memory ima vrijednost od oko 247.000 KB, što uz vrijednosti countera Page life expectancy, Buffer cache hit ratio i Available Mbytes pokazuje da memorije ima dovoljno.

Koliko SQL Server koristi disk možemo vidjeti i preko countera **SQL Server Buffer Manager:Reads/sec** i **Writes/sec**. Oni pokazuju koliko je bilo fizičkih čitanja, odnosno zapisivanja stranica na disk.

Da bismo vidjeli koliko otprilike korisnika i SQL naredbi naredbi proizvodi izmjereno opterećenje, možemo pogledati vrijednosti countera **SQL Server:General Statistics User connections** i **SQL Server:SQL Statistics: Batch requests/sec**. U našem primjeru imamo u prosjeku 3 konekcije koje proizvode oko 35 batcheva (skupine SQL naredbi) u sekundi.

Analiza koju smo dosad proveli bazirala se na vrijednostima countera koje predstavljaju njihov prosjek u cijelom periodu mjerenja. No, korisno je vidjeti i koje su bile maksimalne i minimalne vrijednosti te kako su se one mijenjale.

- Na alatnoj traci kliknite na gumb  (Properties), a zatim se postavite na

	<p>karticu General i odaberite prikaz maksimalnih vrijednosti countera.</p> <ul style="list-style-type: none"> • Pogledajte kako su se promijenile vrijednosti na izvještaju • Podesite sada da se prikazuju minimalne vrijednosti countera i pregledajte ih • Kliknite opet na gumb  i prebacite se na karticu Source. Tamo preko klizača postavite Time Range, tako da se prikazuju counteri iz nekog manjeg podintervala pa ih pregledajte. Zatim ponovite isto to, ali odabirući neke druge intervale. Na kraju se vratite nazad na prikaz cijelog intervala. • Prebacite se sada na grafički prikaz izvještaja: <ul style="list-style-type: none"> ○ Na alatnoj traci kliknite na gumb  i odaberite Line. • Na donjem panelu, u tablici s popisom countera pronađite counter Pages/sec i kliknite na njega. • Na alatnoj traci kliknite na gumb . To će podebljati liniju koja prikazuje kretanje vrijednosti tog performance countera. <p>Tu se vidi da je vrijednost countera Pages/sec bila visoka samo u jednom kraćem periodu na početku mjerenja te da ona prosječna vrijednost ne odražava njezino pravo stanje kroz cijeli period mjerenja.</p> <ul style="list-style-type: none"> • Učinite isto i za neke druge counterere da biste vidjeli kako su se kretale njihove vrijednosti. <p>Možemo primijetiti da postoji jedan interval u kojemu je diskovna aktivnost izrazito pojačana te drugi, u kojem je procesor opterećen do maksimuma. Trebalo bi dodano nadgledati sustav da se utvrdi koji upiti proizvode tu veliku diskovnu, odnosno procesorsku aktivnost.</p>
--	---

Završetak lab

Nakon završetka lab nemojte gasiti virtualni stroj ABP-VM2!

Lab 2: Korištenje Activity Monitora

Priprema za lab

Za ovaj lab treba nastaviti koristiti virtualni stroj ABP-VM2. Nije nužno da je riješen prethodni lab.


Cilj lab

U ovom labu koristit ćete **Activity Monitor** da analizirate procese koji se izvode na instanci SQL Servera. Pogledat ćete ima li blokiranih procesa, na koje resurse procesi čekaju, nad kojim datotekama ima najviše I/O prometa te koji su upiti „najteži“.

Zadaci

Zadaci od kojih se ovaj lab sastoji su:

- Pokretanje i podešavanje Activity Monitora
- Nadgledanje procesa na instanci SQL Servera

Zadatak	Detaljni koraci
1. Pokretanje i podešavanje Activity Monitora	<ul style="list-style-type: none">• Pokrenite Management Studio i logirajte se na instancu ABP-VM2• Na alatnoj traci kliknite gumb  da pokrenete Activity Monitor• Desno kliknite na panel Overview i postavite vrijednost za refresh interval na jednu sekundu
2. Nadgledanje procesa na instanci SQL Servera	<ul style="list-style-type: none">• S alatne trake pokrenite aplikaciju Workloader i odaberite sljedeći scenarij:<ul style="list-style-type: none">○ Upiti za blokiranje: 1○ Upit za veliko opterećenje: 2• Kliknite na gumb Pokreni upite <p>Napomena: Activity Monitor pokazuje što se <i>trenutačno</i> događa na instanci SQL Servera. Ne postoji mogućnost snimanja zabilježenih događaja da bi se kasnije analizirali. Ako neki trenutak želimo detaljnije analizirati, možemo pauzirati snimanje.</p> <ul style="list-style-type: none">• Nakon otprilike jedne minute desno kliknite na panel Overview i kliknite Pause da zaustavite snimanje.• Pregledajte panel Processes<ul style="list-style-type: none">○ Pogledajte je li u trenutku zaustavljanja snimanja bilo blokiranih procesa.<p>Blokirani procesi u koloni Blocked By imaju upisan ID procesa koji ih je blokirao. Upit koji je uzrokovao blokiranje u koloni Head Blocker ima upisano 1.</p><ul style="list-style-type: none">○ Pogledajte koji se upit izvodio pod procesom koji je uzrokovao blokiranje:<ul style="list-style-type: none">▪ Desno kliknite na head blocker proces i odaberite Details. U novom prozoru prikazat će se tekst zadnje SQL naredbe koju je taj proces izveo.<p>Moguće je da je blokiranje prestalo nakon što smo pauzirali snimanje pa da se sada pod tim procesom izvršava neki drugi SQL upit.</p>• Pregledajte panel Resource Waits<ul style="list-style-type: none">○ Pogledajte na koje su resurse procesi najviše čekali.<p>Ako kliknete na naziv neke kolone, podaci će se sortirati.</p>• Pregledajte panel Data File I/O<ul style="list-style-type: none">○ Pogledajte nad kojim je datotekama bilo najviše čitanja i zapisivanja podataka• Pregledajte panel Recent Expensive Queries<ul style="list-style-type: none">○ Pogledajte koji su upiti najviše opterećivali server○ Desno kliknite na jedan od upita i odaberite Edit Query da se upit prikaže u editoru.

Bez obzira koji se refresh interval podesi, panel Recent Expensive Queries osvježava se svakih 30 sekundi i prikazuje upite koji su bili „najteži“ u tom periodu.

Završetak laba

Nakon završetka laba nemojte gasiti virtualni stroj ABP-VM2!

Lab 3: Korištenje Profilera

Priprema za lab

Za ovaj lab treba nastaviti koristiti virtualni stroj ABP-VM2. Nije nužno da je riješen prethodni lab.

Cilj laba


U ovom labu koristit ćete **Profiler** da zabilježite događaje i SQL naredbe koje se izvršavaju na instanci SQL Servera. Podesit ćete da se bilježe detalji o međusobnim blokiranjima procesa te da se filtriraju samo upiti koji dolaze iz određene aplikacije. Na kraju ćete pregledati koji su upiti sudjelovali u blokiranjima i izdvojiti upite koji su nadulje trajali i napravili najviše logičkih čitanja.

Zadaci

Zadaci od kojih se ovaj lab sastoji su:

- Podešavanje postavki i pokretanje *tracea*
- Pregled zabilježenih događaja i SQL upita

Zadatak	Detaljni koraci
1. Podešavanje postavki i pokretanje <i>tracea</i>	<ul style="list-style-type: none">• Pokrenite Management Studio i logirajte se na instancu ABP-VM2• Iz izbornika Tools odaberite SQL Server Profiler• Kod pokretanja Profilera morat ćete se spojiti na instancu SQL Servera koju želite nadgledati. I tu odaberite ABP-VM2• U prozoru Trace properties, na kartici General prihvatite ponuđeni template Standard (default)• Prebacite se na karticu Events Selection• U desnom donjem kutu uključite opciju Show all events• U kategoriji Errors and Warnings odaberite događaj Blocked process report• Da bi se mogli bilježiti detalji o blokiranjima, to najprije treba omogućiti u postavkama instance SQL Servera na sljedeći način:<ul style="list-style-type: none">○ Prebacite se u Management Studio, u Object Exploreru desno kliknite na instancu ABP-VM2 i odaberite Properties○ U Properties prozoru postavite se na karticu Advanced○ Pod Blocked Process Threshold upišite 5 (to znači da će se događaj koji provjerava ima li blokiranih procesa okidati svakih 5 sekundi)• Vratite se nazad u Profiler• Podesite da se hvataju samo upiti koji dolaze iz aplikacije Workloader:<ul style="list-style-type: none">○ Kliknite gumb Column Filters...

	<ul style="list-style-type: none"> ○ Odaberite ApplicationName ○ Pod Like upišite %.NET% i kliknite OK (Workloader će u <i>traceu</i> pod Application name imati upisano <i>.NET SqlClient Data Provider</i>) ● Pokrenite <i>tracing</i> klikom na gumb Run ● Pokrenite aplikaciju Workloader s alatne trake i odaberite sljedeći scenarij: <ul style="list-style-type: none"> ○ Upit za blokiranje: 1 ○ Upit za veliko opterećenje: 2 ● Nakon što aplikacija Workloader završi, pauzirajte trace klikom na gumb  na alatnoj traci Profilera
<p>2. Pregled zabilježenih događaja i SQL upita</p>	<ul style="list-style-type: none"> ● Među zabilježenim događajima pronađite prvu pojavu Blocked process reporta. U gornjem panelu označite redak koji se odnosi na blokiranje, a u donjem panelu pronađite koji upiti u njemu sudjeluju. Koji upit uzrokuje blokiranje, a koji je blokiran? To možete vidjeti u XML tekstu pod tagovima <blocked-process> i <blocking-process> ● Pritisnite CTRL + F i preko prozora Find pronađite naredbu koja u sebi sadrži tekst alter index ● Pronađite sada upit koji je najdulje trajao. Sučelje u Profileru ne omogućava da se zabilježeni upiti sortiraju po nekoj koloni pa je za ovakvu analizu zgodno pospremiti podatke iz <i>tracea</i> u neku tablicu na SQL Serveru: <ul style="list-style-type: none"> ○ U Profileru odaberite File → Save as → Trace Table... ○ Spojite se na instancu ABP-VM2 ○ Pod Database odaberite tempdb, a pod Table upišite TraceTable pa kliknite OK ○ Prebacite se u Management Studio i na instanci ABP-VM2 otvorite bazu tempdb. ○ U bazi tempdb, na popisu tablica desno kliknite na dbo.TraceTable i odaberite opciju Select Top 1000 Rows ○ Doradite upit koji se izgenerirao, tako da na kraj još dodate order by duration desc ○ Upit koji je najduže trajao sada se nalazi u prvom retku rezultata. U tom retku kopirajte podatak iz kolone TextData i napravite paste u novi editor da biste bolje vidjeli njegov kod. ● Pronađite upit koji je proizveo najviše <i>readova</i>: <ul style="list-style-type: none"> ○ Vratite se nazad u prethodni editor i preuredite upit tako da umjesto order by duration desc stavite order by reads desc. Izvedite preuređeni upit ○ Prvi redak ne sadrži upit koji je najduže trajao, nego se on odnosi na događaj Audit Logout i zapravo prikazuje koliko je ukupno readova napravljeno pod jednom od konekcija koju je uspostavio Workloader. Selektirajte zato prvi redak u kojem TextData nije prazan i kopirajte TextData u novi editor da vidite o kojem se upitu radi. <p><i>Reads</i> predstavlja broj logičkih čitanja koje je neki upit napravio. Kad optimiziramo upite, jedan od najvažnijih zadataka je smanjiti broj <i>readova</i>.</p>

Završetak laba

Nakon završetka laba nemojte gasiti virtualni stroj ABP-VM2!

Lab 4: Kreiranje indeksa predloženog u prikazu izvedbenog plana

Priprema za lab

Za ovaj lab nužno je da je riješen lab 3. Treba nastaviti koristiti virtualni stroj ABP-VM2.


Cilj laba

U ovom labu vidjet ćete kako možete zaključiti bi li se neki upit mogao optimizirati kreiranjem indeksa. Kroz Management Studio za određeni upit možemo pogledati kakav je njegov izvedbeni plan. Optimizator upita, koji prikazuje izvedbeni plan, pritom može utvrditi da bi se upit brže izvršavao kad bi postojao neki dodatni indeks. U tom slučaju u izvedbenom planu prikazat će se posebna poruka s naredbom za kreiranje tog indeksa koji nedostaje.

Zadaci

Zadaci od kojih se ovaj lab sastoji su:

- Odabir upita zabilježenog u Profilerovom *traceu*
- Pregled izvedbenog plana za odabrani upit
- Kreiranje predloženog indeksa

Zadatak	Detaljni koraci
1. Odabir upita zabilježenog u Profilerovom <i>traceu</i>	<ul style="list-style-type: none">• Prebacite se u Profiler iz prošle vježbe i među zabilježenim upitima pronađite jedan koji u sebi sadrži tekst SpecialOfferID• Kopirajte njegov tekst• U Management Studiju otvorite novi editor• Postavite se u bazu AdventureWorks2014• Upišite sljedeće naredbe: <pre>Set statistics time on Set statistics io on</pre>• Ovim naredbama daje se instrukcija da se kod izvođenja upita u ispis poruka uvrste podaci o tome koliko je procesorskog vremena te logičkih i fizičkih čitanja bilo utrošeno. Ti će nam podaci biti važni za usporedbu efikasnosti upita prije i nakon kreiranja indeksa.• Sada napravite <i>paste</i> da upit iz Profilera ubacite u editor
2. Pregled izvedbenog plana za odabrani upit	<ul style="list-style-type: none">• Na alatnoj traci kliknite na gumb  (Include Actual Execution Plan). On uključuje opciju za prikaz izvedbenog plana po kojem se upit izvršavao.• Pokrenite sve upite odjednom klikom na gumb Execute• U panelu s rezultatima upita kliknite na karticu Messages

	<p>Primijetite da je u izvođenju upita nad tablicom SalesOrderDetail trebalo napraviti 1238 logičkih čitanja i da se upit izvršavao 227 milisekundi.</p> <ul style="list-style-type: none"> • Kliknite sada na karticu Execution Plan <p>Osim prikaza operatora koji su se koristili u izvedbenom planu i njihovih „cijena“, prikazana nam je i poruka <i>Missing Index</i> s visokim utjecajem na upit od 98.8634.</p>
<p>3. Kreiranje predloženog indeksa</p>	<ul style="list-style-type: none"> • Desno kliknite na poruku Missing index i odaberite opciju Missing Index Details... • Dobivenu naredbu za kreiranje indeksa otkomentirajte i umjesto [<Name of Missing Index, sysname,>] indeksu dajte naziv [IX_SalesOrderDetail_SpecialOfferId] • Izvedite tako preuređenu naredbu za kreiranje indeksa. • Prebacite se u prethodni editor u kojem ste maloprije izveli upit s prikazom execution plana • Ponovno izvedite isti batch, zajedno s prikazom <i>time</i> i <i>io</i> statistike <p>Primijetite da se sada broj logičkih čitanja nad tablicom SalesOrderDetail smanjio na 14, a elapsed time na 49 milisekundi.</p> <p>Ovo je jedan od najjednostavnijih načina za optimiziranje pojedinačnog upita. Nažalost, često ćemo biti u situaciji da svi potrebni indeksi već postoje i da se upit mora optimizirati na neki drugi način. Osim toga, kad optimizator na ovaj način predloži indeks, on ga predlaže imajući u vidu samo taj jedan upit. Kad bismo na taj način optimizirali dva slična upita, moglo bi se dogoditi da optimizator za svakog od njih predloži drugačiji indeks pa mi kreiramo dva indeksa. No, možda se mogao kreirati jedan malo drugačiji indeks koji bi zadovoljavao oba upita. Nikako ne bismo smjeli doći u iskušenje da krenemo optimizirati velik broj upita i da za svakog od njih nekritički kreiramo predloženi missing index. Tako bismo mogli nagomilati indekse i degradirati performanse jer bi bilo potrebno više vremena za njihovo ažuriranje.</p>

Završetak laba

Nakon završetka laba nemojte gasiti virtualni stroj ABP-VM2!

