INFLUENȚA VOLUMULUI DE DATE ASUPRA TIMPULUI DE EXECUȚIE A DIFERITOR TIPURI DE INSTRUCȚIUNI ÎN SQL SERVER

Irina GRIGORAȘ

Departamentul Ingineria Software și Automatică, gr. TI-205, Universitatea Tehnică a Moldovei, Chișinău, Republica Moldova

Autorul corespondent: Grigoraș Irina, irina.grigoras@isa.utm.md

Coordonator științific: Dorian SARANCIUC, DISA, FCIM, UTM

Rezumat. *În articol este descris un experiment asupra categoriilor de instrucțiuni cel mai des utilizate în limbajul SQL. A fost creat un cod de generare a unei baze de date cu 3 tabele. Un tabel a fost populat cu date de 10.000, 100.000, 1.000.000 și 5.000.000, după care secvențial au fost executate interogări simple, fără index, cu index cluster și non-cluster, și s-a verificat timpul de execuție pentru fiecare dintre acestea. Conform rezultatelor s-au demonstrat afirmații din diferite surse despre comenzile SELECT, UPDATE, DELETE.*

Cuvinte cheie: *DDL, DML, DQL, index, cluster, non-cluster, timp de execuție.*

Introducere

În această lucrarea voi încerca să demonstrez informații de pe internet despre timpul de execuție a anumitor comenzi în SQL efectuat în SQL Server. Au fost create tabele pentru proiect cărora le-au fost alocate diferite volume de date, unde a fost măsurat timpul de execuție la cazuri anumite.

În special m-am concentrat pe comenzile DELETE și UPDATE utilizate pentru tabele care conțin sau nu index cluster/non-cluster. Conform datelor obținute am efectuat concluzii și le-am comparat cu surse de pe internet.

1. Instruncțiuni DDL

Limbajul de definire a datelor (DDL, Data Definition Language) este un limbaj orientat la structura bazei de date. DDL permite crearea, modificarea şi suprimarea obiectelor bazei de date (relaţiilor, atributelor, constrângerilor, viziunilor, indecşilor,...). El permite, de asemenea, definirea domeniilor de date (numere, secvenţe de caractere, date temporale.) şi adăugarea constrângerilor asupra valorilor datelor [1].

În cadrul experimentului au fost create 3 tabele conectate între ele pentru o școală auto, un tabel reprezentând contractele clienților ce include id-ul grupelor din care fac parte, iar grupurile conțin id-ul cursului pe care îl urmează grupul dat, exemplu categoria A, B etc.

În tabelul 1 este prezentat timpul de execuție față de numărul de clienți a efectuării instrucțiunilor DROP – pentru ștergerea bazei dacă există, CREATE – crearea tabelelor, ALTER – adăugarea constrângerilor și execuția unor funcții cu WHILE, IF pentru generarea unor date.

Tabelul 1

Crearea tabelelor, dependențelor și generarea datelor cu INSERT

Timp	Numărul de clienți
4 sec	10.000
31 sec	100.000
4 min 33 sec	1.000.000
18 min 46 sec	5.000.000

Conform datelor din Tab. 1 am constatat ca există o legitate de creștere a timpului în dependență de numărul de clienți. Aceasta este că timpul crește cu 8n la numărul clienților înmulțit cu 10, unde n este timpul obținut la calculul anterior.

Tabelul 2

Demonstrații

Nr d/o	Calcul teoretic	Calcul experimental	Concluzie
1	la 10.000 clienți avem timp n = 4 sec, deci la 100.000 timpul ar trebui sa fie 8n = 8*4 = 32 (sec)	31 sec	31 ~ 32 Trecut
2	100.000 clienți => n = 31 sec, la 1.000.000 timpul ar trebui sa fie 8n = 8*31 = 248 (sec)	4 min 33 sec = 273 sec	248 ~ 273 Trecut
3	1.000.000 clienți => n = 273 sec, la 1.000.000 timpul ar trebui sa fie 8n = 8*273 = 2184 (sec)	Consideram că 18 min 46 sec e jumătate și în total va fi 37 min 52 sec = 2272 sec	2184 ~ 2272 Trecut

Deși după cum se vede în Tab. 2 există puțină neconcordanță a datelor, dar asta poate fi provocată de starea calculatorului și capacitatea de executare a interogării. Trebuie de menționat că în cazul dat legitatea de creștere s-a dovedit a fi 8n, în alte cazuri ea poate să diferă acest lucru fiind influnțați de o mulțime de factori.

2. Instruncțiuni DQL

Limbajul de interogare a datelor (DQL, Data Query Language) include instrucţiuni SQL care permit extragerea datelor din baza de date. Este limbajul în care utilizatorii pot formula cereri la baza de date. Deşi reprezintă o parte foarte importantă a limbajului SQL, DQL este format din instrucţiuni care folosesc o singură comandă: SELECT [1].

Timpul de execuție pentru o instrucțiune SELECT cu sau fără JOIN în SQL Server poate varia în funcție de o serie de factori, cum ar fi dimensiunea tabelelor, numărul de rânduri care trebuie preluate, complexitatea interogării și cât de încărcat e serverul.

În Tab. 3 se poate observa timpul de execuție pentru comanda SELECT cu și fără join și clauza WHERE.

Tabelul 3

Comanda SELECT

Timp de execuție (sec)	Cu join și clauza where (sec)	Numărul de clienți
0,35	0,1	10.000
0,44	0,15	100.000
0,50	0,35	1.000.000
59	18	5.000.000

Un motiv de ce datele diferă atât de mult constă în faptul că i-a timp pentru ca fiecare rând să fie afișat, deci întru-cât SELECT-ul simplu afișează toate liniile oricât de multe ar fi, iar cea cu JOIN și clauza WHERE datorită filtrării datelor, economisându-se timp la afișare.

O instrucțiune SELECT fără JOIN se va executa de obicei mai rapid decât o instrucțiune SELECT cu JOIN, mai ales dacă tabelul care este interogat este mare și condiția JOIN implică mai multe tabele. Acest lucru se datorează faptului că atunci când se utilizează un JOIN, SQL Server trebuie să combine rânduri din mai multe tabele, ceea ce poate fi o operație mai complexă și mai consumatoare de timp. Datorită faptului că s-a utilizat și clauza WHERE care a filtrat datele și astfel mai puține date fiind necesar de afișat am obținut acești timpi.

3. Instruncțiuni DML cu indecși cluster și non-cluster

Limbajul de manipulare a datelor (DML, Data Manipulation Language) constituie setul de comenzi referitor la actualizarea conţinutului bazei de date. DML permite introducerea a noi tupluri, modificarea tuplurilor, suprimarea tuplurilor nedorite din baza de date şi blocarea relaţiilor. Instrucţiunile principale sunt INSERT, UPDATE, DELETE, LOCK [1].

În SQL Server, index cluster pe o coloana este creat automat de constrângerea cheii primare. Protocolul specifică că fiecare tabel va avea un singur index cluster. Similar cu modul în care este utilizat un dicționar, un index cluster este utilizat pentru a defini ordinea, sortarea tabelului sau organizarea datelor în ordine alfabetică. Când un tabel are un asemenea index, tabelul este numit tabel cluster. Dacă un tabel nu are index cluster, rândurile sale de date sunt stocate într-o structură neordonată numită heap [2].

Indicii non-cluster au o structură separată de rândurile de date. Un index non-cluster conține valorile cheii de index non-cluster și fiecare intrare de valoare cheie are un pointer către rândul de date care conține valoarea cheie [3].

Tabelul 4

Timpului de execuție a diferitor comenzi SQL pentru diferițe tipuri de indecși

Tip	Update (sec)	Delete (sec)	Numărul de clienți
1. Simplu	0,35	0,25	10.000
	0,94	0,34	100.000
	2,54	2,45	1.000.000
	20,23	15,11	5.000.000
2. Clustered	0,45	0,33	10.000
	0,87	0,47	100.000
	2,20	2,00	1.000.000
	14,46	20,03	5.000.000
3. Nonclustered	0,37	0,27	10.000
	0,40	3,00	100.000
	3,04	9,23	1.000.000
	20,34	47,68	5.000.000

Din Tabelul 4 se poate observa diferențele de comportament a instrucțiunilor UPDATE și DELETE la utilizarea lor în tabel ce nu conține un anumit tip de index și cel ce conține index cluster și non-cluster. După cum se vede la in tabelul ce conține index cluster întrodus prim Primary KEY, trimpul de execuție la ambele comenzi a fost mai rapid decât la acela fără nici un anumit tip de index.

Timpul necesar pentru executarea unei instrucțiuni UPDATE sau DELETE poate varia în funcție de un număr de factori, cum ar fi dimensiunea tabelului, numărul de rânduri afectate de instrucțiune și complexitatea interogării. Referitor la cazul relatat despre execuția mai rapida a comenzii UPDATE față de DELETE de mai sus la primele 2 tipuri, la al treilea tip, non-cluster index, se observă complet opusul, aici execuția la actualizare a luat practic ca și celelalte 2, dar în schimb la ștergere aceasta a luat mai mult timp.

Conform unor surse, executarea instrucțiunii UPDATE poate dura mai mult decât instrucțiunea DELETE dacă tabelul are un număr mare de rânduri și interogarea este complexă, lucru demonstrat în partea din tipul 1 și 2. Acest lucru se datorează faptului că instrucțiunea UPDATE trebuie să citească și să modifice fiecare rând afectat, în timp ce instrucțiunea DELETE trebuie doar să localizeze și să elimine rândurile.

Când se utilizează instrucțiunea de actualizare, SQL Server trebuie să modifice datele din rând și să actualizeze orice indici, ceea ce poate dura mai mult, mai ales dacă tabelul are un număr mare de rânduri sau dacă tabelul are indecși care trebuie actualizați. Pe de altă parte, atunci când se utilizează instrucțiunea de ștergere, SQL Server trebuie să elimine rândurile și să actualizeze indecșii, ceea ce poate fi mai rapid, mai ales dacă tabelul are un număr mare de rânduri, iar instrucțiunea de actualizare trebuie să actualizeze multe rânduri.

Concluzii

În lucrarea dată s-a experimentat timpul de execuție a diferitor comenzi în SQL Server efectuate asupra tabelelor cu diferite volume de date. S-a constatat că la crearea tabelelor s-a conturat o legitate care permite de a prognoza timpul de execuție la mărirea numărului de date inserate. Comanda SELECT, în general, oferă o execuție mai rapidă efectuându-se filtrări a datelor, datorită micșorării numărului de date necesare de a fi afișate. Cel mai lung timp la această comanda este afișarea datelor pentru utilizator.

Altă concluzie care s-a confirmat este că comanda UPDATE cu creșterea datelor incetinește mai rapid decât DELETE, astfel că aceasta are o execuție mai avantajoasă pentru aceleași date. În general, timpul de execuție obținut de SQL SERVER diferă de la un calculator la altul, și în timpul experimentării s-au obținut timpi diferiți. Am demonstrat infomații obținute din internet despre indecși cluster și non-cluster, și anume că la DELETE indexul non-cluster se execută mai greu decât cel cluster sau chiar fără. Deci, trebuie de decis cum avem de gând să utilizăm un tabel ca mai apoi să decidem ce instrucțiune sau index să utilizăm.

Referințe

1. V. COTELEA, *Algebra relațională și limbajul SQL*. Chișinău: Vizual Design, 2013.
2. Diference between Clustered and Non-clustered Index [online]. [accesat 26.02.2023]. Disponibil: https://byjus.com/gate/difference-between-clustered-and-non-clustered-index/
3. Clustered and nonclustered indexes described [online]. [accesat 01.02.2023]. Disponibil: https://learn.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver16