

Indexarea tabelelor

1. Obiectivul lucrării:

Formarea și dezvoltarea abilităților de interogare a bazelor de date.

2. Breviar teoretic cu exerciții și probleme rezolvate

Un index este o structură opțională a bazei de date care permite accesarea directă a unui rând dintr-un tabel. Indecșii pot fi creați pentru una sau mai multe coloane a unui tabel, în acest ultim caz folosindu-se denumirea de *indecși compuși* sau *indecși concatenați*. Un index este utilizat de către sistemul de gestiune al bazei de date pentru a găsi rapid valori pentru coloana sau coloanele pentru care a fost creat indexul, furnizând o cale de acces directă la rândurile asociate acestora fără a mai fi necesară investigarea fiecărui rând din tabel. Practic, în momentul în care se dorește căutarea anumitor înregistrări ale căror valori îndeplinesc un anumit criteriu, în loc să se parcurgă tabelul, se parcurge indexul, acesta din urmă furnizând localizarea exactă a înregistrărilor ce îndeplinesc criteriul de căutare.

Prezența indecșilor este transparentă pentru utilizator și aplicație, SQL Server ia decizia utilizării unui index pe baza unor date care sintetizează distribuția informațiilor din tabele, prin urmare indecșii nu trebuie obligatoriu să fie referiți în mod direct prin interogări. În plus, sintaxa unei comenzi SQL nu este în nici un fel influențată de existența indecșilor, iar rezultatele oricărei interogări vor fi aceleași indiferent dacă există indecși sau nu.

Indecșii sunt independenți din punct de vedere fizic și logic de datele din tabelul de bază. Un index poate fi creat și distrus fără ca datele din tabelul de bază sau ceilalți indecși să aibă de suferit. Singurul lucru pe care indexul îl va modifica va fi durata accesului la datele tabelului, acesta devenind mai lent în absența indexului. Evident, odată cu ștergerea unui tabel sunt șterși și indecșii asociați acestuia.

Indecșii pot să fie *unici* sau *ne-unici*. Indecșii unici garantează faptul că nu va exista nici o pereche de rânduri care să aibă valori identice pentru coloana sau grupul de coloane pentru care a fost definit indexul. Valorile *Null* sunt ignorate de criteriul de unicitate. Un rând cu valoarea *Null* în coloana indexată nu va fi înregistrat în index, deci un index unic nu va împiedica stocarea mai multor rânduri cu o valoare *Null* în coloana indexată. Un index ne-unic nu impune nicio restricție în legătură cu valorile din coloanele care îl definesc.

Odată definit, un index este actualizat de către baza de date ori de câte ori au loc modificări ale datelor tabelului. Aceasta înseamnă că ori de câte ori au loc inserări, ștergeri sau modificări ale datelor unui tabel, toți indecșii acelui tabel sunt actualizați în mod automat, acest lucru având ca efect încetinirea acestor operații asupra datelor tabelului. Cu alte cuvinte, existența indecșilor sporește viteza accesului la datele tabelului pe de o parte, dar în același timp încetinește operațiile de modificare ale acestora. De aceea, nu trebuie creat un index pentru fiecare coloană a unui tabel (deși acest lucru este posibil), ci pentru anumite coloane cheie ale acestuia. De exemplu, se recomandă indexarea coloanelor care conțin în majoritate valori unice sau un domeniu larg de valori sau coloane după care se fac dese căutări sau ordonări.

Deoarece prezența indecșilor poate avea un impact semnificativ asupra eficienței aplicației în continuare prezentăm câteva sugestii privind folosirea acestora¹:

Ce tabele trebuie indexate:

¹ Florentin Eugen Ipat, Monica Popescu, Dezvoltarea aplicațiilor de baze de date în Oracle 8 și Forms 6

- Indexați tabelele pentru care majoritatea interogărilor selectează doar un număr redus de rânduri (sub 5%). Interogările care selectează un număr mare de rânduri nu folosesc în mod eficient indecșii.
- Nu indexați tabele ce conțin puține înregistrări deoarece în acest caz accesul secvențial va fi mai rapid.
- Indexați tabelele care sunt interogate folosind fraze SQL simple. Clauzele SQL mai complexe nu folosesc cu aceeași eficiență indecșii.
- Nu indexați tabelele care sunt actualizate frecvent. Inserările, modificările și ștergerile sunt îngreunate de existența indecșilor. Decizia de a indexa un tabel trebuie luată pe baza raportului dintre numărul de interogări și cel de actualizări efectuat asupra acestuia.
- Indexați tabelele care nu au valori duplicate în coloanele care apar în clauza `WHERE` a celor mai frecvente interogări.

Ce coloane trebuie indexate:

- Folosiți coloanele cele mai frecvent folosite în clauza `WHERE` a interogărilor.
- Nu se recomandă indexarea coloanelor cu număr mic de valori distincte.
- Coloanele care au valori distincte sunt candidate foarte bune pentru indexare. De altfel, SQL Server creează în mod automat indecși unici pentru coloanele definite ca `PRIMARY KEY` sau `UNIQUE`. În plus, din punct de vedere logic, este preferabil ca indecșii unici să nu fie definiți în mod explicit, ci prin intermediul constrângerilor `PRIMARY KEY` și `UNIQUE` - acesta deoarece unicitatea este un concept logic și ar trebui să fie asociat cu definiția tabelului.
- Coloanele care sunt folosite pentru a face legătura dintre tabele sunt în general candidate pentru indexare. În general, se recomandă indexarea cheilor străine.
- În anumite situații, folosirea indecșilor compuși poate fi mai eficientă decât a celor individuali. De exemplu, crearea indecșilor compuși este recomandabilă când două coloane nu sunt unice fiecare în parte dar combinația lor este unică sau are în majoritate valori unice. De asemenea, se recomandă crearea indecșilor compuși atunci când interogările uzuale ale tabelului conțin în clauza `WHERE` mai multe coloane individuale legate prin `AND`.

Indecșii consumă spațiu în baza de date la fel ca și tabelele și există într-un spațiu tabel la fel ca și acestea. Spațiul necesar indecșilor pentru tabele mari este de obicei semnificativ, așa că trebuie planificat din momentul în care se proiectează baza de date.

În SQL un index se creează folosind comanda `CREATE INDEX`. O sintaxă simplificată a acestei comenzi se prezintă astfel:

```
CREATE [unique] [clustered | nonclustered] INDEX denumire_index
ON tabel (coloana [ASC|DESC] [, ...])
```

unde:

- Dacă este specificată opțiunea `UNIQUE`, indexul creat conține valori unice (ceea ce implică faptul că în tabel nu pot exista două rânduri cu valori identice în coloanele ce formează indexul), altfel nu.
- *Clustered* indică faptul că pentru indexul creat, ordinea fizică a tupelor coincide cu cea logică corespunzătoare indexului. Aceasta implică reordonarea fizică a rândurilor tabelului. Pentru un tabel se poate defini cel mult un index `clustered` la un moment dat. Valoarea implicită este *nonclustered*.

Exemple:

```
Create index Ix_CNP on tStudenti(CNP)
```

```
create index Ix_NumeComplet on tStudenti( nume, prenume)
```

```
Select * from tStudenti where CNP='1910517123456'  
-- parcurgere dupa indexul Ix_CNP
```

```
Select * from tStudenti  
where Nume='Radu' and prenume='Ion' -- parcurgere dupa indexul Ix_NumeComplet
```

```
Select * from tStudenti where Nume='Radu'; -- parcurgere dupa indexul  
Ix_NumeComplet
```

Crearea indecșilor unui tabel se recomandă să se facă după ce tabelul a fost populat cu date - aceasta deoarece existența indecșilor încetinește în mod evident inserarea datelor.

La definirea constrângerilor PRIMARY KEY și UNIQUE sau la activarea acestora, SQL Server creează în mod automat indecși unici pentru coloanele sau grupurile de coloane respective. În acest caz denumirea indexului coincide cu denumirea constrângerii.

Impunerea utilizării unui anumit index.

SQL Server permite referirea unui index într-o frază SELECT prin specificarea acestuia în clauza FROM. Sintaxa simplificată este:

```
SELECT lista_de_coloane FROM denumire_tabel WITH (INDEX=denumire_index)
```

sau

```
SELECT lista_de_coloane FROM denumire_tabel WITH (INDEX(denumire_index) )
```

Exemple:

```
Select * from tStudenti with (index=ixNumeComplet)
```

```
Select * from tStudenti as A with (index=ixNumeComplet)  
inner join tRezultateExamene as B on A.CodStd=B.CodStd
```

Un index poate fi eliminat folosind comanda DROP INDEX cu următoarea sintaxă:

```
DROP INDEX DenumireTabel.denumire_index;
```

De exemplu:

```
DROP INDEX tStudenti.IxNumeComplet;
```

Un index nu poate fi șters dacă el a fost creat în mod automat, ca parte a definirii sau a activării unei restricții PRIMARY KEY sau UNIQUE. În acest caz indexul este șters automat la ștergerea sau dezactivarea constrângerii.

Ștergerea unui index se face de obicei dacă indexul nu mai este necesar, sau înainte de încărcarea masivă a datelor într-un tabel; ștergerea unui index înainte de încărcarea masivă a datelor și recrearea lui după terminarea încărcării va duce la îmbunătățirea performanței încărcării precum și la utilizarea mai eficientă a spațiului alocat indexului.