

INDECȘI și utilizarea lor în optimizarea accesului la date

I. OBIECTIVE

1. Conceptul de index
2. Sintaxa MySQL pentru crearea și utilizarea indecșilor
3. Strategii de creare a indecșilor

II. FUNDAMENTE TEORETICE

2.1.CREAREA ȘI UTILIZAREA INDECȘILOR

Una dintre cele mai importante responsabilități ale proiectantului unei baze de date este să *definiească tabelele (relațiile) astfel încât performanțele să fie optime*. Structura de bază a unei tabeli nu specifică modul în care datele urmează să fie accesate sau stocate fizic, ci doar *constrângerile* impuse de tipurile de date și diversele constrângeri asupra datelor, la nivel de coloane sau tabel (chei primare/chei străine, integritatea referențială).

MySQL oferă un *mecanism de indexare a tabelor care facilitează optimizarea răspunsului la interogări diverse asupra datelor*. În lipsa indecșilor, MySQL ar trebui să efectueze parcurgeri diverse și posibil multiple ale tabeli, adică să citească fiecare linie a unei tabeli pentru a răspunde la o interogare. În cazul tabelor foarte mari, această abordare este în mod evident una costisitoare pentru server și generează timpi de așteptare a răspunsului nedoriți. În accepțiune nonformală, indecșii reprezintă o modalitate prin care MySQL *organizează pointeri de acces* la datele solicitate.

O definiție mai apropiată implementării afirmă faptul că un index dintr-o bază de date reprezintă o structură ce conține *o listă de valori semnificative, împreună cu referințe la intrările din tabela bazei de date* în care se află informațiile corespunzătoare. Indecșii reprezintă metoda internă prin care MySQL organizează accesul la datele din tabelă, astfel încât acestea să poată fi regăsite în mod optim (timp minim). Concret, un index este un obiect în baza de date ce necesită spațiu de stocare, iar avantajul oferit este acela că reduce numărul de operații de acces la disc, necesare pentru regăsirea unei anumite informații dintr-o tabelă.

2.1.1.Crearea indecșilor

Crearea indecșilor în MySQL se poate realiza cu comanda CREATE INDEX, care are sintaxa următoare:

```
CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX index_name
  [index_type]
  ON tbl_name (key_part,...)
  [index_option]
  [algorithm_option | Lock_option] ...
```

```
key_part:
  col_name [(length)] [ASC | DESC]
```

```
index_option: {
  KEY_BLOCK_SIZE [=] value
  | index_type
```

```
| WITH_PARSER parser_name  
| COMMENT 'string'  
}
```

```
index_type:  
    USING {BTREE | HASH}
```

```
algorithm_option:  
    ALGORITHM [=] {DEFAULT | INPLACE | COPY}
```

```
lock_option:  
    LOCK [=] {DEFAULT | NONE | SHARED | EXCLUSIVE}
```

Unde:

- nume_index – numele indexului trebuie să fie unic în baza de date. □ tabela – tabela de indexat.
- coloana – coloana utilizată în indexare.
- UNIQUE înseamnă MySQL va crea o constrângere ca toate valorile în index trebuie să fie distincte.
- FULLTEXT este acceptat numai în coloanele care au tipul de date CHAR, VARCHAR sau TEXT.
- Indexul SPATIAL (pentru date spațiale)este disponibil pentru motoarele de baze de date InnoDB, MyISAM, NDB.

Exemple pentru crearea diverselor tipuri de indecși și diverse alte detalii pot fi consultate la adresele :

<https://dev.mysql.com/doc/refman/5.7/en/create-index.html>

<http://www.mysqltutorial.org/mysql-index>

https://www.w3schools.com/sql/sql_create_index.asp

Observații:

1. Indecșii sunt **logic și fizic** (pentru cele mai multe SGBD-uri) **independenți** de bazele de date pe care le indexează. Acest lucru înseamnă că indecșii pot fi creați sau șterși **oricând**, fără nici un efect asupra bazelor de date sau a altor indecși.
2. Când este șters un tabel, indecșii corespunzători sunt de asemenea șterși.
3. Indecșii pot fi creați **explicit** sau **automat** (vezi index pe cheie primară în MySQL). Indexul explicit este *automat folosit și întreținut de către sistem*, astfel odată creat, nu este necesară nici o intervenție din partea utilizatorului.
4. Pentru coloanele de tip șir de caractere, MySQL dă posibilitatea indexării *pe o anumită parte din șirul de caractere*. Ca un exemplu, putem să considerăm numele dintr-un tabel cu date despre studenții dintr-o facultate. Considerăm deasemenea că tipul de date pentru câmpul nume este varchar(20). În această situație indexarea după primele 10 caractere ale numelui nu ar afecta prea mult timpul de căutare comparativ cu o indexare după primele 20 de caractere, iar avantajul ar fi salvarea spațiului pe hard disk prin crearea unui index mult mai mic.

Regulile generale în căutarea unor **buni candidați** pentru un index sunt:

1. în cazul interogărilor care returnează o **mulțime rezultat** de tuple **mare**, este foarte utilă indexarea.
2. **cheile străine** reprezintă candidați de bază pentru indecși, pentru cazul în care datele sunt extrase folosind operatorul JOIN sau corelarea după cheia străină pentru accesarea conținutului din două tabele

3. indexarea după coloanele specificate în clauzele **Order By și Group By** aduc deasemenea avantaje sub aspectul timpului de regăsire a datelor.
4. Fiecare index compozit(creat pe baza mai multor attribute) posedă anumite restricții legat de lungimea și numărul de coloane, astfel lungimea maximă a unui index este 767 bytes, și poate conține max 16 coloane.
5. MySQL ia în calcul utilizarea unui index numai dacă în instrucțiunea SELECT se specifică aceasta, în caz contrar, indexul este pur și simplu ignorat.

Detalii suplimentare referitor la limitări impuse de MySQL în crearea indecșilor pot fi consultate la adresa : <https://dev.mysql.com/doc/refman/5.6/en/innodb-restrictions.html>.

2.1.2.Ștergerea indecșilor

Eliminarea din sistem a indecșilor se realizează cu comanda:

```
DROP INDEX index_name ON tbl_name
[algorithm_option | lock_option] ...
```

algorithm_option:

```
ALGORITHM [=] {DEFAULT | INPLACE | COPY}
```

lock_option:

```
LOCK [=] {DEFAULT | NONE | SHARED | EXCLUSIVE}
```

Instrucțiunea șterge indexul cu numele *index_name* din tabela *table_name*, fiind mapată la o comandă ALTER TABLE. Inplace este algoritmul implicit utilizat.Lock_option controlează nivelul de citiri/scrieri concurente în baza de date .

Diverse exemple de utilizare pot fi consultate la adresa

<https://www.mysqltutorial.org/mysql-index/mysql-drop-index/>

2.1.3. Vizualizarea indecșilor

Pentru vizualizarea indecșilor creați într-o tabelă se va utiliza comanda

```
SHOW INDEXES FROM tbl_name
```

Pentru vizualizarea indecșilor creați în baza de date ,se va utiliza comanda

```
SHOW INDEXES FROM tbl_name IN database_name;
```

Detalii și exemple pot fi consultate la adresa

<https://www.mysqltutorial.org/mysql-index/mysql-show-indexes/>

Pentru afișarea tuturor indecșilor din baza de date se poate utiliza interogarea :

```
select distinct
  table_name,
  index_name
from information_schema.statistics
where table_schema = 'sakila';
```

Respectiv, pentru afișarea tuturor atributelor se poate utiliza interogarea :

```
Select index_schema,  
       index_name,  
       group_concat(column_name order by seq_in_index) as index_columns,  
       index_type,  
       case non_unique  
         when 1 then 'Not Unique'  
         else 'Unique'  
       end as is_unique,  
       table_name  
from information_schema.statistics  
where table_schema not in ('information_schema', 'mysql',  
                           'performance_schema', 'sys')  
  
group by index_schema,  
         index_name,  
         index_type,  
         non_unique,  
         table_name  
order by index_schema, index_name;
```

2.1.4 Utilizarea indecșilor

Pentru utilizarea indecșilor, se va utiliza comanda USE INDEX integrate într-o fraza SQL validă.

SELECT listă, FROM tabel USE INDEX (listă_indecși) WHERE condiție.

2.2.STRATEGII DE CREARE A INDECȘILOR

Deși SGBD-ul nu posedă nici un fel de limitări practice în ceea ce privește numărul de indecși suportați, este de dorit limitarea acestora din următoarele motive : de ex. aceștia *utilizează spațiu disc* fiind astfel posibilă creșterea dimensiunii indecșilor peste dimensiunea bazei de date. Deasemenea prin contrast cu beneficiile utilizării indecșilor la regăsirea datelor, pentru operații de inserare și actualizare indecșii activi au un impact negativ, deoarece *creșterea numărului de indecși activi ai unei tabele determină creșterea numărului de operații de reorganizare* a acestora în cazul unor operații pe date.

Este recomandat crearea unui index să fie realizată pentru următoarele situații:

- un câmp ce este utilizat des în *clauzele Where* sau în *conditii join*
- câmpuri cu mare varietate de valori
- câmpuri cu număr mare de valori NULL
- două sau mai multe câmpuri ce sunt folosite împreună frecvent într-o clauza WHERE sau într-o condiție join (index compus)
- baza de date este mare și majoritatea interogărilor nu vizează mai mult de 2-4% din înregistrări

Mai mulți indecși într-o bază de date *nu înseamnă în mod automat o optimizare a interogărilor*. Fiecare operație de manipulare a datelor ce se realizează într-o bază de date ce conține indecși , implică o actualizare a indecșilor. Cu cât sunt mai mulți indecși asociați tabelului care trebuie astfel menținuti, cu atât va dura mai mult până când serverul va actualiza indecșii .

Nu este necesară crearea unui index atunci când:

- baza de date este mică
- câmpurile nu sunt des folosite într-o condiție
- majoritatea interogărilor vizează aprox. 2-4% din înregistrări

Indecși și condiții în clauza WHERE. Dacă clauza WHERE conține o condiție cu o singură coloană , aceasta se va constitui ca și coloană de indexare. Dacă în clauză apar mai mulți operatori AND, este necesară crearea unui index compozit ce include *toate coloanele specificate*.

Indecși și operatorul JOIN. În situația existenței unui operator JOIN, este recomandată indexarea fiecărei coloane JOIN, acestea de regulă reprezintă cheia primară a unei tabele , respectiv cheia străină a celeilalte,O creștere semnificativă de performanță este realizată dacă toate coloanele din interogare sunt cuprinse în index.

III. ACTIVITĂȚI PRACTICE

3.1. Definiți pe scurt următoarele tipuri de indecsi : prefix,unique, invizibil, compozit, clusterat

3.2.Se va implementa în Workbench baza de date *Sakila* pusă la dispozitie la adresa <https://dev.mysql.com/doc/indexother.html> și se vor identifica indecșii asociați ,analizând proprietățile lor.

3.3.Se vor executa interogări notând timpii de răspuns în condiții de execuție cu și fără utilizarea indecșilor.