



СУБД: Индекси

Пламен Николов, СУ ФМИ

Резюме

- ✓ Какво представляват **индексите**?
- ✓ Кога трябва да се използват **индекси**?
- ✓ Как се създават/редактират/трият **индекси**?
- ✓ Примери

Какво представляват индексите?

Индексът в СУБД е:

- ✓ Обект в базата от данни
- ✓ Използва се от СУБД за да ускори намирането на редове(кортежи), като за целта използва указатели
- ✓ Може да намали дисковите Входно/Изходнит операции, като използва бърз метод за намиране на местоположението на данните
- ✓ Прилага се към атрибутите на една релация
- ✓ Използва се и се поддържа автоматично от СУБД

Използване на индекси

Кога трябва да се използват индекси:

- Атрибутът(колоната) съдържа голям брой различни стойности
- Един или повече атрибути(колони) са често използвани заедно в WHERE клауза(съпоставяни с константи) или в условие за свързване
- Таблицата е голяма и повечето заявки се очаква да върнат по-малко от 20% от редовете.

Повече не винаги значи по-добре!

Повече индексирание съвсем не означава по-бързи заявки. При почти всяка DML операция върху таблица с индекси индексите трябва да бъдат променени. Колкото повече индекси имаме върху таблицата, толкова повече усилия се изискват от СУБД, за да обнови индексите след DML операцията.

Използване на индекси

Пример:

```
SELECT *  
FROM Movie  
WHERE studioName = 'Disney' AND year = 1990;
```

Ако Movie съдържа 10 000 реда и само 200 имат стойност за year = 1990, то по-ефективния начин е да се търси кои от тези 200 филма има studioName = 'Disney', а не от всичките 10 000 записа.

Използване на индекси

Кога не трябва да се дефинират индекси:

- ✓ Релацията е сравнително малка по обем съхранявани данни
- ✓ Атрибутите не се използват често в условия в заявките
- ✓ Повечето заявки се очаква да върнат повече от 20% от кортежите в релацията
- ✓ Данните в таблицата се променят често
- ✓ Индексираните колони се използват като част от израз, а не самостоятелно (в този случай може да е удачно да се създаде т.нар. функционален индекс)

Дефиниране на индекси

Индексите могат да бъдат създавани **експлицитно** или **автоматично**. Ако няма индекс върху дадена колона, тогава за намиране на дадена стойност в колоната се осъществява пълно претърсване на таблицата. Индексът осигурява бърз и директен достъп до редовете от таблицата. Той е физически независим от таблицата, която индексира. Под независимост се има предвид, че индекси могат да бъдат създавани и изтривани по всяко време без тези операции да оказват влияние върху базовите таблици или другите индекси върху тях. Когато изтриете таблица обаче, съответните индекси също биват изтрити.

Дефиниране на индекси

Автоматично:

Уникален индекс се създава автоматично, когато се дефинира първичен и уникален ключ върху колоните на дадена таблица. Повечето СУБД поддържат тази функционалност с цел полесна поддръжка на съответните ограничения за цялостност. По отношение на уникалните ключове обаче е добре да се провери в документацията на конкретната СУБД, дали автоматично се създава уникален индекс за съответната колона/колони.

Дефиниране на индекси

Експлицитно създаване на индекс

```
CREATE [UNIQUE] INDEX index_name  
ON table(column[,column]...)
```

В конкретния случай:

```
CREATE INDEX YearIndex ON Movie(year)
```

Дефиниране на индекси

Уникален индекс може да се създаде експлицитно като се зададе ключовата дума `UNIQUE`. Такъв индекс не допуска наличието на две еднакви стойности в съответната колона/комбинация от колони. Препоръчва се обаче да създавате уникалните индекси имплицитно (неявно), като зададете съответното `PRIMARY KEY / UNIQUE` ограничение върху колоните на таблицата.

Пример:

```
CREATE INDEX idx_ships_classes  
ON ships (class);
```

Премахване на индекс

Изтриването на индекс се осъществява с оператора:

```
DROP INDEX table_name.index_name;
```

Обикновено СУБД не предоставя средства за модифициране на индекси. Ако искате да модифицирате индекс, то трябва първо да го изтриете, а после да го пресъздадете.

Пример

Нека имаме следната релация:

StarsIn(movieTitle, movieyear, starName)

Изпълняваме предимно три вида заявки към тази релация – **Q1**, **Q2** и **I**:

**Q1: SELECT movieTitle, movieYear
FROM StarsIn
WHERE starName = S;**

**Q2: SELECT starName
FROM StarsIn
WHERE movieTitle = t AND movieYear = y;**

Пример

I: **INSERT INTO StarsIn VALUES(t, y, s);**

Нека времето което е необходимо за обхождането на **StarsIn** го оценим с **10**(зависи от това на колко дискови блокове е разположена, в случая 10)

Приемаме, че една филмова звезда играе в средно 3 филма и един филм има средно 3 звезди.

При така зададените данни имаме, че при наличието на индекс на **starName** или на комбинацията на **movieTitle** и **movieYear** оценката на времето за достъп ще е **3**(3 пъти достъп до диска). Без индекси оценката остава **10**.

Пример

I: **INSERT INTO StarsIn VALUES(t, y, s);**

Нека времето което е необходимо за обхождането на **StarsIn** го оценим с **10**(зависи от това на колко дискови блокове е разположена, в случая 10)

Когато съпоставим различните случаи получаваме следната таблица:

Действие	Без индекси	Star Индекс	Movie Индекс	И двата са индекс
Q1	10	4	10	4
Q2	10	10	4	4
I	2	4	4	6
	$2 + 8p_1 + 8p_2$	$4 + 6p_2$	$4 + 6p_1$	$6 - 2p_1 - 2p_2$

Реализация на индексите

Когато се говори за индекс в базите данни, без да изрично да се уточнява неговия вид, обикновено се има предвид т.нар. B-tree индекс. Това е индекс, за чиято реализация се използва структурата от данни B дърво (напрактика се използват модификации на B дърво: B*, B+ и др). Листата на B дърво представляват двусвързан подреден списък от стойностите на индексиранията колона/колони. Ако освен стойността на колоната, във възлите на дървото имаме и указател към място от файла с данни, където се намират останалите полета от съответния ред на таблицата, говорим за не-клъстериран индекс. Ако във възлите на дървото се съхраняват целите редове, то индексът се нарича клъстериран. По този начин всички данни от таблицата са изнесени в индекса.

??



Въпроси?