Bazy danych

12. SQL — Wyszukiwanie pełnotekstowe

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

semestr letni 2007/08

Wyszukiwanie pełnotekstowe

Wyszukiwanie pełnotekstowe służy do wyszukiwania podanych napisów (ciągów znaków) w kolumnach typu `CHAR`, `VARCHAR`, `TEXT`. W tym celu na tabeli, którą chcemy w tens posób przeszukiwać, należy założyć specjalny indeks. W MySQL wyszukiwanie pełnotekstowe działa tylko na tabelach typu MyISAM.

```
mysql> CREATE TABLE Teksty
    -> (Id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    -> Tytul VARCHAR(32),
    -> Tresc VARCHAR(64),
    -> FULLTEXT(Tytul,Tresc)
    -> ) ENGINE=MyISAM, CHARSET cp1250;
Query OK, 0 rows affected (0.28 sec)
```

Niech przykładowa tabela zawiera następujące dane:

```
mysql> SELECT * FROM Teksty;
+----+----------------------+-----------------------------------------------------+
| Id | Tytul                | Tresc                                               |
+----+----------------------+-----------------------------------------------------+
|  1 | Prof. Andrzej bim    | Rzepliński, bom, kryminolog: - Nie znam akt          |
|  2 | operacyjnych bim bom | gdzie mogą być dowody, ale nie możemy               |
|  3 | wykluczyć,           | że XXX bim był milicyjnym agentem.                  |
|  4 | Powtórka             | Operacyjnych akt nie znam ja, bim, mówi Rzepliński. |
|  5 | Bim                  | Operacyjnych, operacyjnych, operacyjnych, milicyjych. |
|  6 | Bim bim              | akt operacyjnych                                    |
|  7 | Bom                  | bim                                                 |
|  8 | wypełniacz           | wypełniacz                                          |
|  9 | bim                  | wypełniacz                                          |
| 10 | wypełniacz           | bim                                                 |
+----+----------------------+-----------------------------------------------------+
10 rows in set (0.10 sec)
```

Wyszukiwanie odbywa się poprzez wywołanie zapytania `SELECT` z klauzulą `WHERE MATCH(`*nazwy kolumn*`) AGAINST(`*wzorzec*`)`.

```
mysql> SELECT * FROM TEKSTY
    -> WHERE MATCH (Tytul,Tresc) AGAINST('operacyjnych');
+----+----------------------+-----------------------------------------------------+
| Id | Tytul                | Tresc                                               |
+----+----------------------+-----------------------------------------------------+
|  5 | Bim                  | Operacyjnych, operacyjnych, operacyjnych, milicyjnych. |
|  6 | Bim bim              | akt operacyjnych                                    |
|  4 | Powtórka             | Operacyjnych akt nie znam ja, bim, mówi Rzepliński. |
|  2 | operacyjnych bim bom | gdzie mogą być dowody, ale nie możemy               |
+----+----------------------+-----------------------------------------------------+
4 rows in set (0.00 sec)
```

Wartość semantyczna

Wyrażenie `MATCH...AGAINST(`*wzorzec*`)` określa wartość semantyczną podanego wzorca w poszczególnych wierszach tabeli:

```
mysql> SELECT Id, CONCAT(Tytul,' ',Tresc) AS Tekst,
    -> MATCH (Tytul,Tresc) AGAINST('operacyjnych') AS Wartosc
    -> FROM Teksty;
+----+-------------------------------------------------------------+------------------+
| Id | Tekst                                                       | Wartosc          |
+----+-------------------------------------------------------------+------------------+
|  1 | Prof. Andrzej bim Rzepliński, bom, kryminolog: - Nie znam akt |      0           |
|  2 | operacyjnych bim bom gdzie mogą być dowody, ale nie możemy  | 0.38341854994499 |
|  3 | wykluczyć, że XXX bim był milicyjnym agentem.               |      0           |
|  4 | Powtórka Operacyjnych akt nie znam ja, bim, mówi Rzepliński. | 0.38763393589171 |
|  5 | Bim Operacyjnych, operacyjnych, operacyjnych, milicyjnych.  | 0.53687456835829 |
|  6 | Bim bim akt operacyjnych                                    | 0.40085528270084 |
|  7 | Bom bim                                                     |      0           |
|  8 | wypełniacz wypełniacz                                       |      0           |
|  9 | bim wypełniacz                                              |      0           |
| 10 | wypełniacz bim                                              |      0           |
+----+-------------------------------------------------------------+------------------+
10 rows in set (0.00 sec)
```

Wartość semantyczna wzorca w wierszu

- jest tym większa, im więcej razy wzorzec występuje w wierszu,
- jest tym większa, im wiersz jest krótszy,
- przyjmuje wartość zero, jeśli wzorzec *nie* występuje w wierszu.

Uwaga: Wartość semantyczna wyrazów krótkich lub występujących w co najmniej połowie wierszy wynosi *zero*.

```
mysql> SELECT * FROM TEKSTY
    -> WHERE MATCH (Tytul,Tresc) AGAINST('bim bom');
Empty set (0.00 sec)
```

Zapytanie `SELECT... WHERE MATCH... AGAINST...` wyświetla tylko wiersze o niezerowej wartości semantycznej.

Boolowskie wyszukiwanie pełnotekstowe

`AGAINST(`*wzorzec* `IN BOOLEAN MODE)` pozwala na stosowanie operatorów lepiej określających poszukiwany wzorzec. Przy wyszukiwaniu boolowskim tabela *nie musi* mieć zdefiniowanego indeksu `FULLTEXT`.

- $+$ — fragment musi być obecny, $-$ — fragment nie może być obecny.

```
mysql> SELECT * FROM TEKSTY
    -> WHERE MATCH (Tytul,Tresc) AGAINST('+operacyjnych -znam' IN BOOLEAN MODE);
+----+----------------------+-----------------------------------------------------+
| Id | Tytul                | Tresc                                               |
+----+----------------------+-----------------------------------------------------+
|  2 | operacyjnych bim bom | gdzie mogą być dowody, ale nie możemy               |
|  5 | Bim                  | Operacyjnych, operacyjnych, operacyjnych, milicyjnych. |
|  6 | Bim bim              | akt operacyjnych                                    |
+----+----------------------+-----------------------------------------------------+
3 rows in set (0.04 sec)
```

- Brak operatora — podany fragment jest opcjonalny. Wiersze zawierające fragment opcjonalny są oceniane wyżej.

- () — nawiasy służą do grupowania słów w podwyrażenia. Takie grupy można zagnieżdżać.
- " — zwrot ujęty w podwójne cudzysłowy powoduje dopasowanie tylko wierszy, które zawierają podany zwrot *dokładnie w tej formie, w jakiej został napisany*.

Są także inne operatory.

```
mysql> SELECT * FROM Teksty
    -> WHERE MATCH(Tytul,Tresc) AGAINST('+(akt operacyjnych)' IN BOOLEAN MODE);
+----+----------------------+-----------------------------------------------------+
| Id | Tytul                | Tresc                                               |
+----+----------------------+-----------------------------------------------------+
|  2 | operacyjnych bim bom | gdzie mogą być dowody, ale nie możemy               |
|  4 | Powtórka             | Operacyjnych akt nie znam ja, bim, mówi Rzepliński. |
|  5 | Bim                  | Operacyjnych, operacyjnych, operacyjnych, milicyjnych. |
|  6 | Bim bim              | akt operacyjnych                                    |
+----+----------------------+-----------------------------------------------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM Teksty
    -> WHERE MATCH(Tytul,Tresc) AGAINST('+"akt operacyjnych"' IN BOOLEAN MODE);
+----+---------+------------------+
| Id | Tytul   | Tresc            |
+----+---------+------------------+
|  6 | Bim bim | akt operacyjnych |
+----+---------+------------------+
1 row in set (0.00 sec)
```

Wyszukiwanie z rozwijaniem zapytania

Użytkownik bardzo często polega na “wiedzy niejawnej” — człowiek wie, że jakieś terminy są równoznaczne lub bliskoznaczne, ale jak to zalgorytmizować? Powiedzmy, szukając wzorca “daza danych”, chcemy też uzyskać wiersze zawierające napisy “Oracle” i “MySQL”, nawet jeśli napis “baza danych” w nich nie występuje. Albo też szukając informacji o profesorze, chcemy także znaleźć wiersze, w których profesor wymieniony jest z nazwiska, z pominięciem tytułu.

Realizujemy to za pomocą konstrukcji `AGAINST(`*wzorzec* `WITH QUERY EXPANSION)`. W takim wypadku tabela przeszukiwana jest dwa razy. Przy drugim wyszukiwaniu wzorzec zostaje połączony z kilkoma innymi wzorcami, znajdującymi w wierszach będących na górze (mających największą wartość semantyczną *pierwotnego wzorca*) po pierwszym wyszukiwaniu.

Przykład

```
mysql> SELECT * FROM TEKSTY
    -> WHERE MATCH (Tytul,Tresc) AGAINST('Prof');
+----+------------------+-------------------------------------------+
| Id | Tytul            | Tresc                                     |
+----+------------------+-------------------------------------------+
|  1 | Prof. Andrzej bim | Rzepliński, bom, kryminolog: - Nie znam akt |
+----+------------------+-------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM TEKSTY
    -> WHERE MATCH (Tytul,Tresc) AGAINST('Prof' WITH QUERY EXPANSION);
+----+------------------+-------------------------------------------------+
| Id | Tytul            | Tresc                                           |
+----+------------------+-------------------------------------------------+
|  1 | Prof. Andrzej bim | Rzepliński, bom, kryminolog: - Nie znam akt      |
|  4 | Powtórka         | Operacyjnych akt nie znam ja, bim, mówi Rzepliński. |
+----+------------------+-------------------------------------------------+
2 rows in set (0.01 sec)
```

Uwagi końcowe

- Wyszukiwanie pełnotekstowe jest wolne i kosztowne.
- Jeżeli planujemy przeszukiwać naprawdę dużą tabelę, zaleca się
  - Utworzyć tabelę bez definiowania indeksu `FULLTEX`.
  - Wprowadzić dane do tabeli.
  - Po wprowadzeniu danych odać indeks za pomocą polecenia `ALTER TABLE`.

  W przeciwnym wypadku wprowadzanie danych do tabeli może być bardzo powolne.
- Istnieje sporo *komercyjnych* nakładek na RDBMSy, znacznie usprawniających wyszukiwanie pełnotekstowe.