Многомерни и битови индекси. Дървовидни структури за многомерни данни в MySQL

Валентина Динкова, ф.н.71112

ФМИ

3 юни 2010 г.

GIS и разширението на MySQL за пространствени данни

- Какво е **GIS** и какво е **OGC**?

GIS означава Географска Информационна Система и е един от най-очевидните примери за пространствени данни.

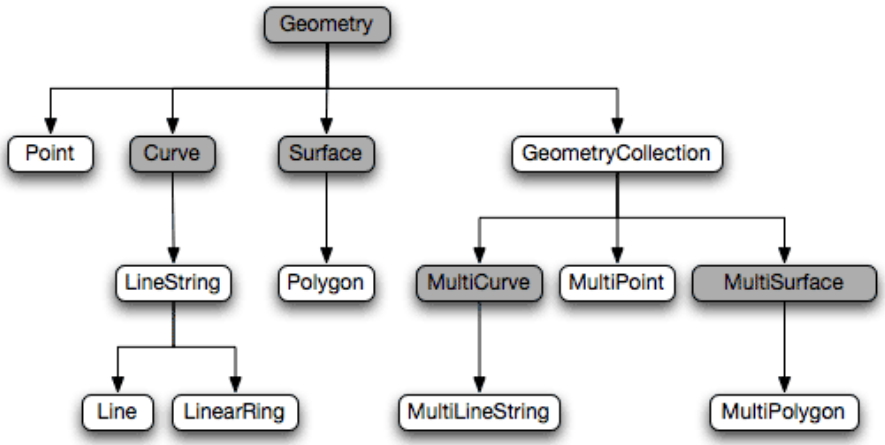
OGC (Open Geospatial Consorcium) е организация, която работи по стандартизирането на различни области на GIS. Един такъв стандарт е и спецификацията за SQL, която определя разширението на SQL базирани релационни бази данни, което да използва GIS обекти и операции.

OGC работи в 4 важни области:

- типове данни;
- операции;
- възможност да се подават като вход и да се извеждат GIS данни;
- индексиране на пространствени данни.

Друга важна област са метаданните.

Стандартът, използван от почти всички SQL бази данни с пространствено разширение, включително и MySQL



Типовете, отбелязани в сиво са абстрактни и обекти от тези типове не могат да се създават.

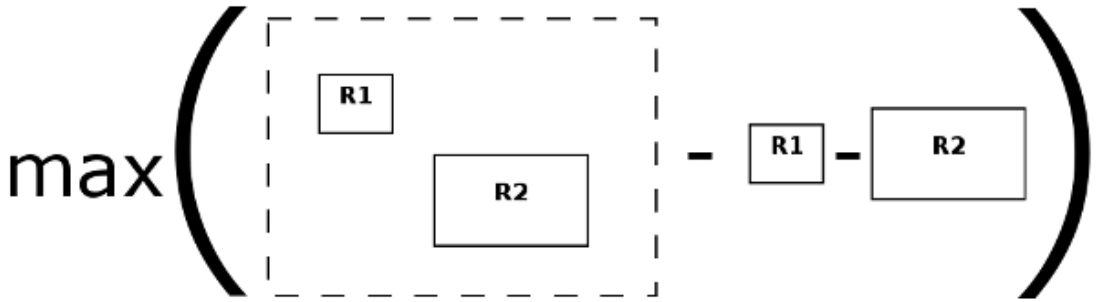
Пространствени индекси

- Пространствените данни могат да се индексират също както останалите данни в MySQL. Но за да бъде индексирането ефективно, се използва пространствен тип индексиране, реализирано чрез R-дървета. MySQL използва R-дървета с квадратично разделяне;
- Не всички engine-и поддържат многомерни индекси.

R-дървета с квадратично разделяне

Добавяме нов запис. Нека сме намерили листото, където трябва да добавим новия запис и нека $M =$ *"брой региони в листо"*

- Избираме 2 от $M + 1$ записа да бъдат първите елементи на двете нови листа, като избираме двойката,която би заела най-много място ако и двата елемента се постават на едно място (двойката при която покриващия регион ще е най-голям). Намираме тази двойка като от областта покриваща двата записа изваждаме самите записи и искаме тази разлика да е най-голяма.



- Останалите записи разделяме в двете листа един по един. На всяка стъпка разширяването, необходимо за добавянето на всеки от оставащите записи към всяко листо се изчислява и добавеният запис е този, който е показал най-голяма разлика спрямо двете листа.

Създаваме таблицата *map_test*, където *loc* е пространствен атрибут

```
mysql> create table map_test
    -> (
    ->   name varchar(100) not null primary key,
    ->   loc  geometry not null,
    -> );
Query OK, 0 rows affected (0.00 sec)
```

Чрез следната процедурата добавяме 30 000 реда

```
mysql> CREATE PROCEDURE fill_points(IN size INT(10))
    -> BEGIN
    ->   DECLARE i DOUBLE(10,1) DEFAULT size;
    ->
    ->   DECLARE lon FLOAT(7,4);
    ->   DECLARE lat FLOAT(6,4);
    ->   DECLARE position VARCHAR(100);
    ->
    ->   DELETE FROM map_test;
    ->
    ->   WHILE i > 0 DO
    ->     SET lon = RAND() * 360 - 180;
    ->     SET lat = RAND() * 180 - 90;
    ->
    ->     SET position = CONCAT( 'POINT(', lon, ' ', lat, ')' );
    ->
    ->     INSERT INTO map_test(name, loc) VALUES ( CONCAT('name_', i), GeomFromText(position) );
    ->
    ->     SET i = i - 1;
    -> END WHILE;
    -> END @
Query OK, 0 rows affected (0.08 sec)

mysql> delimiter ;
mysql> CALL fill_points(30000);
Query OK, 1 row affected (26.00 sec)
```

Заявка за проверка кои точки се съдържат в полигона

```
mysql> SELECT name, AsText(loc) FROM map_test WHERE
    -> Contains(
    -> GeomFromText('POLYGON((0 0, 0 1, 1 1, 2 0, 0 0))'),
    -> loc) = 1;

+--------------+----------------------+
| name         | AsText(loc)          |
+--------------+----------------------+
| name_12768.0 | POINT(0.6646 0.7551) |
| name_14707.0 | POINT(1.8941 0.1449) |
| name_29530.0 | POINT(0.1938 0.0931) |
+--------------+----------------------+
3 rows in set (0.12 sec)
```

Сега създаваме пространствен индекс по атрибута *loc*

```
mysql> create spatial index ps_index on map_test(loc);
Query OK, 30000 rows affected (2.44 sec)
Records: 30000  Duplicates: 0  Warnings: 0
```

И отново правим същата заявка

```
mysql> SELECT name, AsText(loc) FROM map_test WHERE
    -> Contains(
    -> GeomFromText('POLYGON((0 0, 0 1, 1 1, 2 0, 0 0))'),
    -> loc) = 1;

+--------------+----------------------+
| name         | AsText(loc)          |
+--------------+----------------------+
| name_12768.0 | POINT(0.6646 0.7551) |
| name_14707.0 | POINT(1.8941 0.1449) |
| name_29530.0 | POINT(0.1938 0.0931) |
+--------------+----------------------+
3 rows in set (0.05 sec)
```