**ỦY BAN NHÂN DÂN TP.HCM**

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**



**BÁO CÁO ĐỒ ÁN CHUYÊN ĐỀ WEB 1**

**NĂM HỌC 2018 - 2019**

**Tên đề tài: Tìm hiểu cơ sở dữ liệu MySQL**

**Phần chỉ mục Index**

**GV: Phan Thanh Nhuần**

**SVTH:**

**Lê Minh Thuận (16211TT1805)**

TP. Hồ Chí Minh, năm 2019

1. **Báo nhiệm vụ cá nhân:**
2. **Tìm hiểu chỉ mục Fulltext**

FULLTEXTcác chỉ mục được tạo trên các cột dựa trên văn bản ( [CHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html), [VARCHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html)hoặc [TEXT](https://dev.mysql.com/doc/refman/5.7/en/blob.html)cột) để giúp tăng tốc các truy vấn và hoạt động DML trên dữ liệu chứa trong các cột đó, bỏ qua mọi từ được định nghĩa là từ khóa.

Một FULLTEXTchỉ mục được định nghĩa là một phần của [CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html)câu lệnh hoặc được thêm vào bảng hiện có bằng cách sử dụng [ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html) hoặc [CREATE INDEX](https://dev.mysql.com/doc/refman/5.7/en/create-index.html).

Tìm kiếm toàn văn được thực hiện bằng [MATCH() ... AGAINST](https://dev.mysql.com/doc/refman/5.7/en/fulltext-search.html#function_match)cú pháp.

InnoDB FULLTEXT các chỉ mục được mô tả theo các chủ đề sau trong phần này:

1. [**Thiết kế chỉ mục toàn văn của InnoDB**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-design)**.**
2. [**Bảng chỉ mục toàn văn của InnoDB**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-tables)**.**
3. [**Bộ đệm ẩn chỉ mục toàn văn bản của InnoDB**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-cache)**.**
4. [**ID tài liệu chỉ mục toàn văn bản của InnoDB và cột FTS\_DOC\_ID**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-docid)**.**
5. [**InnoDB Xử lý xóa chỉ mục toàn văn bản**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-deletion)**.**
6. [**InnoDB Xử lý giao dịch chỉ mục toàn văn bản**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-transaction)**.**
7. [**Theo dõi chỉ mục toàn văn của InnoDB**](https://dev.mysql.com/doc/refman/5.7/en/innodb-fulltext-index.html#innodb-fulltext-index-monitoring)**.**
8. **Thiết kế chỉ mục toàn văn của InnoDB**

InnoDB FULLTEXTcác chỉ mục có một thiết kế chỉ mục đảo ngược. Các chỉ mục đảo ngược lưu trữ một danh sách các từ và đối với mỗi từ, một danh sách các tài liệu mà từ đó xuất hiện. Để hỗ trợ tìm kiếm gần, thông tin vị trí cho mỗi từ cũng được lưu trữ, dưới dạng bù byte.

1. **Bảng chỉ mục toàn văn của InnoDB**

Khi tạo một InnoDB FULLTEXTchỉ mục, một tập hợp các bảng chỉ mục được tạo, như trong ví dụ sau:

mysql> CREATE TABLE opening\_lines (

id INT UNSIGNED AUTO\_INCREMENT NOT NULL PRIMARY KEY,

opening\_line TEXT(500),

author VARCHAR(200),

title VARCHAR(200),

FULLTEXT idx (opening\_line)

) ENGINE=InnoDB;

mysql> SELECT table\_id, name, space from INFORMATION\_SCHEMA.INNODB\_SYS\_TABLES

WHERE name LIKE 'test/%';

+----------+----------------------------------------------------+-------+

| table\_id | name | space |

+----------+----------------------------------------------------+-------+

| 333 | test/FTS\_0000000000000147\_00000000000001c9\_INDEX\_1 | 289 |

| 334 | test/FTS\_0000000000000147\_00000000000001c9\_INDEX\_2 | 290 |

| 335 | test/FTS\_0000000000000147\_00000000000001c9\_INDEX\_3 | 291 |

| 336 | test/FTS\_0000000000000147\_00000000000001c9\_INDEX\_4 | 292 |

| 337 | test/FTS\_0000000000000147\_00000000000001c9\_INDEX\_5 | 293 |

| 338 | test/FTS\_0000000000000147\_00000000000001c9\_INDEX\_6 | 294 |

| 330 | test/FTS\_0000000000000147\_BEING\_DELETED | 286 |

| 331 | test/FTS\_0000000000000147\_BEING\_DELETED\_CACHE | 287 |

| 332 | test/FTS\_0000000000000147\_CONFIG | 288 |

| 328 | test/FTS\_0000000000000147\_DELETED | 284 |

| 329 | test/FTS\_0000000000000147\_DELETED\_CACHE | 285 |

| 327 | test/opening\_lines | 283 |

+----------+----------------------------------------------------+-------+

Sáu bảng đầu tiên biểu thị chỉ mục đảo ngược và được gọi là bảng chỉ mục phụ. Khi văn bản đến được tokenized, từ cá nhân (hay còn gọi là “ thẻ ” ) được chèn vào các bảng chỉ số cùng với thông tin vị trí và các tài liệu liên quan đến ID ( DOC\_ID). Các từ được sắp xếp và phân vùng đầy đủ trong số sáu bảng chỉ mục dựa trên trọng số sắp xếp của bộ ký tự của ký tự đầu tiên của từ.

Chỉ mục đảo ngược được phân vùng thành sáu bảng chỉ mục phụ trợ để hỗ trợ tạo chỉ mục song song. Theo mặc định, hai luồng token hóa, sắp xếp và chèn các từ và dữ liệu liên quan vào các bảng chỉ mục. Số lượng chủ đề có thể cấu hình bằng cách sử dụng[innodb\_ft\_sort\_pll\_degree](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_ft_sort_pll_degree) tùy chọn. Xem xét tăng số lượng chủ đề khi tạo FULLTEXTchỉ mục trên các bảng lớn.

Tên bảng chỉ mục phụ được thêm tiền tố FTS\_và tiền tố với INDEX\_\*. Mỗi bảng chỉ mục được liên kết với bảng được lập chỉ mục bởi một giá trị hex trong tên bảng chỉ mục khớp với table\_idbảng được lập chỉ mục. Ví dụ, table\_idcác test/opening\_linesbảng là327, mà giá trị hex là 0x147. Như đã trình bày trong ví dụ trước, các “ 147 ” giá trị hex xuất hiện trong tên của các bảng chỉ số có liên quan đến các test/opening\_linesbảng.

Một giá trị hex đại diện index\_idcủa FULLTEXTchỉ số cũng xuất hiện trong tên bảng chỉ số phụ. Ví dụ: trong tên bảng phụtest/FTS\_0000000000000147\_00000000000001c9\_INDEX\_1, giá trị hex 1c9có giá trị thập phân là 457. Chỉ số được xác định trên opening\_linesbảng ( idx) có thể được xác định bằng cách truy vấn [INFORMATION\_SCHEMA.INNODB\_SYS\_INDEXES](https://dev.mysql.com/doc/refman/5.7/en/innodb-sys-indexes-table.html) bảng cho giá trị này (457).

mysql> SELECT index\_id, name, table\_id, space from INFORMATION\_SCHEMA.INNODB\_SYS\_INDEXES

WHERE index\_id=457;

+----------+------+----------+-------+

| index\_id | name | table\_id | space |

+----------+------+----------+-------+

| 457 | idx | 327 | 283 |

+----------+------+----------+-------+

Các bảng chỉ mục được lưu trữ trong không gian bảng riêng của chúng nếu bảng chính được tạo trong không gian bảng [mỗi tệp](https://dev.mysql.com/doc/refman/5.7/en/glossary.html#glos_file_per_table) .

Các bảng chỉ mục khác được hiển thị trong ví dụ trước được gọi là các bảng chỉ mục chung và được sử dụng để xử lý xóa và lưu trữ trạng thái bên trong của các FULLTEXTchỉ mục. Không giống như các bảng chỉ mục đảo ngược, được tạo cho mỗi chỉ mục toàn văn bản, bộ bảng này là chung cho tất cả các chỉ mục toàn văn bản được tạo trên một bảng cụ thể.

Các bảng phụ trợ thông thường được giữ lại ngay cả khi các chỉ mục toàn văn bản bị loại bỏ. Khi một chỉ mục toàn văn bản bị loại bỏ,FTS\_DOC\_IDcột được tạo cho chỉ mục sẽ được giữ lại, vì việc loại bỏ FTS\_DOC\_ID cột sẽ yêu cầu xây dựng lại bảng. Các bảng nách thông thường được yêu cầu để quản lý FTS\_DOC\_ID cột.

* FTS\_\*\_DELETED và FTS\_\*\_DELETED\_CACHE

Chứa ID tài liệu (DOC\_ID) cho các tài liệu đã bị xóa nhưng dữ liệu chưa được xóa khỏi chỉ mục toàn văn. Đây FTS\_\*\_DELETED\_CACHElà phiên bản trong bộ nhớ của FTS\_\*\_DELETED bảng.

* FTS\_\*\_BEING\_DELETED và FTS\_\*\_BEING\_DELETED\_CACHE

Chứa ID tài liệu (DOC\_ID) cho các tài liệu bị xóa và dữ liệu hiện đang trong quá trình xóa khỏi chỉ mục toàn văn. CácFTS\_\*\_BEING\_DELETED\_CACHEbảng là phiên bản trong bộ nhớ của FTS\_\*\_BEING\_DELETEDbảng.

* FTS\_\*\_CONFIG

Lưu trữ thông tin về trạng thái nội bộ của FULLTEXTchỉ mục. Quan trọng nhất, nó lưu trữ FTS\_SYNCED\_DOC\_ID, trong đó xác định các tài liệu đã được phân tích cú pháp và tuôn ra đĩa. Trong trường hợp khôi phục sự cố, FTS\_SYNCED\_DOC\_IDcác giá trị được sử dụng để xác định các tài liệu chưa được xóa vào đĩa để các tài liệu có thể được phân tích lại và thêm lại vào FULLTEXTbộ đệm chỉ mục. Để xem dữ liệu trong bảng này, truy vấn [INFORMATION\_SCHEMA.INNODB\_FT\_CONFIG](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-config-table.html) bảng.

### **Bộ đệm ẩn chỉ mục toàn văn bản của InnoDB**

Khi một tài liệu được chèn, nó được mã hóa và các từ riêng lẻ và dữ liệu liên quan được chèn vào FULLTEXTchỉ mục. Quá trình này, ngay cả đối với các tài liệu nhỏ, có thể dẫn đến nhiều lần chèn nhỏ vào các bảng chỉ mục phụ, khiến việc truy cập đồng thời vào các bảng này trở thành một điểm gây tranh cãi. Để tránh sự cố này, hãy InnoDBsử dụng FULLTEXT bộ đệm chỉ mục để tạm thời chèn vào bảng chỉ mục cho các hàng được chèn gần đây. Cấu trúc bộ đệm trong bộ nhớ này giữ các phần chèn thêm cho đến khi bộ đệm đầy và sau đó hàng loạt xóa chúng vào đĩa (vào các bảng chỉ mục phụ). Bạn có thể truy vấn[INFORMATION\_SCHEMA.INNODB\_FT\_INDEX\_CACHE](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-index-cache-table.html) bảng để xem dữ liệu mã thông báo cho các hàng được chèn gần đây.

Hành vi xóa bộ đệm và hàng loạt tránh cập nhật thường xuyên cho các bảng chỉ mục phụ, điều này có thể dẫn đến các vấn đề truy cập đồng thời trong thời gian chèn và cập nhật bận rộn. Kỹ thuật tạo khối cũng tránh được nhiều lần chèn cho cùng một từ và giảm thiểu các mục trùng lặp. Thay vì xóa từng từ riêng lẻ, các phần chèn vào của cùng một từ được hợp nhất và được xóa vào đĩa dưới dạng một mục duy nhất, cải thiện hiệu quả chèn trong khi giữ các bảng chỉ mục phụ càng nhỏ càng tốt.

Các [innodb\_ft\_cache\_size](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_ft_cache_size) biến được sử dụng để cấu hình toàn văn kích thước chỉ mục cache (chỉ trên một cơ sở cho mỗi bảng), ảnh hưởng đến mức độ thường xuyên chỉ số bộ nhớ cache toàn văn là đỏ ửng. Bạn cũng có thể xác định giới hạn kích thước bộ đệm chỉ mục toàn văn bản toàn cầu cho tất cả các bảng trong một trường hợp cụ thể bằng cách sử dụng [innodb\_ft\_total\_cache\_size](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_ft_total_cache_size)tùy chọn.

Bộ đệm chỉ mục toàn văn lưu trữ thông tin giống như bảng chỉ mục phụ. Tuy nhiên, bộ đệm chỉ mục toàn văn bản chỉ lưu trữ dữ liệu được mã hóa cho các hàng được chèn gần đây. Dữ liệu đã được xóa vào đĩa (vào các bảng phụ trợ toàn văn bản) sẽ không được đưa trở lại vào bộ đệm chỉ mục toàn văn bản khi được truy vấn. Dữ liệu trong các bảng chỉ mục phụ được truy vấn trực tiếp và kết quả từ các bảng chỉ mục phụ được hợp nhất với các kết quả từ bộ đệm chỉ mục toàn văn bản trước khi được trả về.

### **d. ID tài liệu chỉ mục toàn văn bản của InnoDB và cột FTS\_DOC\_ID**

InnoDBsử dụng một định danh tài liệu duy nhất được gọi là ID tài liệu ( DOC\_ID) để ánh xạ các từ trong chỉ mục toàn văn bản thành các bản ghi tài liệu nơi từ đó xuất hiện. Ánh xạ yêu cầu một FTS\_DOC\_ID cột trên bảng được lập chỉ mục. Nếu một FTS\_DOC\_ID cột không được xác định, InnoDBsẽ tự động thêm một FTS\_DOC\_IDcột ẩn khi chỉ mục toàn văn bản được tạo. Ví dụ sau đây thể hiện hành vi này.

Định nghĩa bảng sau không bao gồm một FTS\_DOC\_IDcột:

mysql> CREATE TABLE opening\_lines (

id INT UNSIGNED AUTO\_INCREMENT NOT NULL PRIMARY KEY,

opening\_line TEXT(500),

author VARCHAR(200),

title VARCHAR(200)

) ENGINE=InnoDB;

Khi bạn tạo một chỉ mục toàn văn bản trên bảng bằng CREATE FULLTEXT INDEXcú pháp, một cảnh báo được trả về sẽ báo cáo việc InnoDBxây dựng lại bảng để thêm FTS\_DOC\_ID cột.

mysql> CREATE FULLTEXT INDEX idx ON opening\_lines(opening\_line);

Query OK, 0 rows affected, 1 warning (0.19 sec)

Records: 0 Duplicates: 0 Warnings: 1

mysql> SHOW WARNINGS;

+---------+------+--------------------------------------------------+

| Level | Code | Message |

+---------+------+--------------------------------------------------+

| Warning | 124 | InnoDB rebuilding table to add column FTS\_DOC\_ID |

+---------+------+--------------------------------------------------+

Cảnh báo tương tự được trả về khi sử dụng [ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html)để thêm chỉ mục toàn văn vào bảng không có FTS\_DOC\_IDcột. Nếu bạn tạo một chỉ mục toàn văn tại [CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html)thời điểm và không chỉ định một FTS\_DOC\_IDcột, hãy InnoDBthêm một FTS\_DOC\_IDcột ẩn , không có cảnh báo.

Xác định một FTS\_DOC\_IDcột tại [CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html)thời điểm ít tốn kém hơn so với việc tạo một chỉ mục toàn văn bản trên một bảng đã được tải dữ liệu. Nếu một FTS\_DOC\_ID cột được xác định trên một bảng trước khi tải dữ liệu, bảng và các chỉ mục của nó không phải được xây dựng lại để thêm cột mới. Nếu bạn không quan tâm đến CREATE FULLTEXT INDEXhiệu suất, hãy bỏ qua FTS\_DOC\_IDcột đểInnoDBtạo nó cho bạn. InnoDBtạo một FTS\_DOC\_IDcột ẩn cùng với một chỉ mục duy nhất ( FTS\_DOC\_ID\_INDEX) trên FTS\_DOC\_IDcột. Nếu bạn muốn tạo FTS\_DOC\_IDcột của riêng mình , cột phải được xác định là BIGINT UNSIGNED NOT NULLvà được đặt tênFTS\_DOC\_ID (tất cả chữ hoa), như trong ví dụ sau:

chú thích

Các FTS\_DOC\_IDcột không cần phải được định nghĩa là một AUTO\_INCREMENTcột, nhưng AUTO\_INCREMENTcó thể làm cho tải dữ liệu dễ dàng hơn.

mysql> CREATE TABLE opening\_lines (

FTS\_DOC\_ID BIGINT UNSIGNED AUTO\_INCREMENT NOT NULL PRIMARY KEY,

opening\_line TEXT(500),

author VARCHAR(200),

title VARCHAR(200)

) ENGINE=InnoDB;

Nếu bạn chọn tự xác định FTS\_DOC\_IDcột, bạn có trách nhiệm quản lý cột để tránh các giá trị trống hoặc trùng lặp. FTS\_DOC\_IDcác giá trị không thể được sử dụng lại, có nghĩa là FTS\_DOC\_ID các giá trị phải ngày càng tăng.

Tùy chọn, bạn có thể tạo duy nhất yêu cầu FTS\_DOC\_ID\_INDEX(tất cả chữ hoa) trên FTS\_DOC\_IDcột.

mysql> CREATE UNIQUE INDEX FTS\_DOC\_ID\_INDEX on opening\_lines(FTS\_DOC\_ID);

Nếu bạn không tạo FTS\_DOC\_ID\_INDEX, hãy InnoDBtạo nó tự động.

Trước MySQL 5.7.13, khoảng cách được phép giữa FTS\_DOC\_IDgiá trị được sử dụng lớn nhất và FTS\_DOC\_IDgiá trị mới là 10000. Trong MySQL 5.7.13 trở lên, khoảng cách được phép là 65535.

Để tránh xây dựng lại bảng, FTS\_DOC\_ID cột được giữ lại khi thả chỉ mục toàn văn.

### **InnoDB Xử lý xóa chỉ mục toàn văn bản**

Xóa một bản ghi có cột chỉ mục toàn văn bản có thể dẫn đến nhiều lần xóa nhỏ trong các bảng chỉ mục phụ, khiến việc truy cập đồng thời vào các bảng này trở thành một điểm gây tranh cãi. Để tránh sự cố này, ID Tài liệu ( DOC\_ID) của tài liệu đã xóa được ghi vào một FTS\_\*\_DELETEDbảng đặc biệt mỗi khi bản ghi bị xóa khỏi bảng được lập chỉ mục và bản ghi được lập chỉ mục vẫn ở trong chỉ mục toàn văn. Trước khi trả về kết quả truy vấn, thông tin trongFTS\_\*\_DELETEDbảng được sử dụng để lọc ID tài liệu đã xóa. Lợi ích của thiết kế này là việc xóa rất nhanh và không tốn kém. Hạn chế là kích thước của chỉ mục không được giảm ngay lập tức sau khi xóa hồ sơ. Để xóa các mục chỉ mục toàn văn bản cho các bản ghi bị xóa, hãy chạy OPTIMIZE TABLEtrên bảng được lập chỉ mục[innodb\_optimize\_fulltext\_only=ON](https://dev.mysql.com/doc/refman/5.7/en/innodb-parameters.html#sysvar_innodb_optimize_fulltext_only) để xây dựng lại chỉ mục toàn văn bản. Để biết thêm thông tin, hãy xem [Tối ưu hóa Chỉ mục Toàn văn của InnoDB](https://dev.mysql.com/doc/refman/5.7/en/fulltext-fine-tuning.html#fulltext-optimize) .

### **InnoDB Xử lý giao dịch chỉ mục toàn văn bản**

InnoDB FULLTEXTcác chỉ mục có các đặc điểm xử lý giao dịch đặc biệt do bộ nhớ đệm và hành vi xử lý hàng loạt của nó. Cụ thể, các cập nhật và chèn vào FULLTEXTchỉ mục được xử lý tại thời điểm cam kết giao dịch, điều đó có nghĩa là FULLTEXTtìm kiếm chỉ có thể thấy dữ liệu đã cam kết. Ví dụ sau đây thể hiện hành vi này. Các FULLTEXTtìm kiếm chỉ trả về kết quả sau khi các dòng chèn được thực hiện.

mysql> CREATE TABLE opening\_lines (

id INT UNSIGNED AUTO\_INCREMENT NOT NULL PRIMARY KEY,

opening\_line TEXT(500),

author VARCHAR(200),

title VARCHAR(200),

FULLTEXT idx (opening\_line)

) ENGINE=InnoDB;

mysql> BEGIN;

mysql> INSERT INTO opening\_lines(opening\_line,author,title) VALUES

('Call me Ishmael.','Herman Melville','Moby-Dick'),

('A screaming comes across the sky.','Thomas Pynchon','Gravity\'s Rainbow'),

('I am an invisible man.','Ralph Ellison','Invisible Man'),

('Where now? Who now? When now?','Samuel Beckett','The Unnamable'),

('It was love at first sight.','Joseph Heller','Catch-22'),

('All this happened, more or less.','Kurt Vonnegut','Slaughterhouse-Five'),

('Mrs. Dalloway said she would buy the flowers herself.','Virginia Woolf','Mrs. Dalloway'),

('It was a pleasure to burn.','Ray Bradbury','Fahrenheit 451');

mysql> SELECT COUNT(\*) FROM opening\_lines WHERE MATCH(opening\_line) AGAINST('Ishmael');

+----------+

| COUNT(\*) |

+----------+

| 0 |

+----------+

mysql> COMMIT;

mysql> SELECT COUNT(\*) FROM opening\_lines WHERE MATCH(opening\_line) AGAINST('Ishmael');

+----------+

| COUNT(\*) |

+----------+

| 1 |

+----------+

### **Theo dõi chỉ mục toàn văn của InnoDB**

Bạn có thể theo dõi và kiểm tra các khía cạnh xử lý văn bản đặc biệt của các InnoDB FULLTEXTchỉ mục bằng cách truy vấn các INFORMATION\_SCHEMA bảng sau :

* [INNODB\_FT\_CONFIG](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-config-table.html)
* [INNODB\_FT\_INDEX\_TABLE](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-index-table-table.html)
* [INNODB\_FT\_INDEX\_CACHE](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-index-cache-table.html)
* [INNODB\_FT\_DEFAULT\_STOPWORD](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-default-stopword-table.html)
* [INNODB\_FT\_DELETED](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-deleted-table.html)
* [INNODB\_FT\_BEING\_DELETED](https://dev.mysql.com/doc/refman/5.7/en/innodb-ft-being-deleted-table.html)

Bạn cũng có thể xem thông tin cơ bản cho FULLTEXTcác chỉ mục và bảng bằng cách truy vấn [INNODB\_SYS\_INDEXES](https://dev.mysql.com/doc/refman/5.7/en/innodb-sys-indexes-table.html)và[INNODB\_SYS\_TABLES](https://dev.mysql.com/doc/refman/5.7/en/innodb-sys-tables-table.html).

1. **Tìm hiểu chỉ mục Spatial:**
2. **Giới thiệu về Spatial**

MySQL có các kiểu dữ liệu không gian tương ứng với các lớp OpenGIS. Cơ sở cho các loại này được mô tả trong [Phần 11.5.2, Mô hình Hình học OpenGIS](https://dev.mysql.com/doc/refman/5.7/en/opengis-geometry-model.html) .

Một số kiểu dữ liệu không gian chứa các giá trị hình học đơn:

* GEOMETRY
* POINT
* LINESTRING
* POLYGON

GEOMETRYcó thể lưu trữ các giá trị hình học của bất kỳ loại. Các loại đơn giá trị khác ( POINT, LINESTRINGvà POLYGON) hạn chế giá trị của họ cho một loại hình cụ thể.

Các kiểu dữ liệu không gian khác chứa các tập hợp các giá trị:

MULTIPOINT

MULTILINESTRING

MULTIPOLYGON

GEOMETRYCOLLECTION

GEOMETRYCOLLECTIONcó thể lưu trữ một bộ sưu tập các đối tượng thuộc bất kỳ loại nào. Các loại bộ sưu tập khác ( MULTIPOINT,MULTILINESTRINGvà MULTIPOLYGON) giới hạn các thành viên bộ sưu tập đối với những người có loại hình học cụ thể.

Ví dụ: Để tạo một bảng có tên geomcột gcó thể lưu trữ các giá trị của bất kỳ loại hình học nào, hãy sử dụng câu lệnh sau:

CREATE TABLE geom (g GEOMETRY);

SPATIALcác chỉ mục có thể được tạo trên NOT NULLcác cột không gian, vì vậy nếu bạn có kế hoạch lập chỉ mục cho cột, hãy khai báo nó NOT NULL:

CREATE TABLE geom (g GEOMETRY NOT NULL);

1. **Cách tạo chỉ mục Spatial**

Đối với InnoDBvà MyISAM bảng, MySQL có thể tạo các chỉ mục không gian bằng cách sử dụng cú pháp tương tự như để tạo các chỉ mục thông thường, nhưng sử dụng SPATIALtừ khóa. Các cột trong chỉ mục không gian phải được khai báo NOT NULL. Các ví dụ sau minh họa cách tạo các chỉ mục không gian:

Với [CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html):

CREATE TABLE geom (g GEOMETRY NOT NULL, SPATIAL INDEX(g));

Với [ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html):

CREATE TABLE geom (g GEOMETRY NOT NULL);

ALTER TABLE geom ADD SPATIAL INDEX(g);

Với [CREATE INDEX](https://dev.mysql.com/doc/refman/5.7/en/create-index.html):

CREATE TABLE geom (g GEOMETRY NOT NULL);

CREATE SPATIAL INDEX g ON geom (g);

SPATIAL INDEX tạo ra một chỉ số cây R. Đối với các công cụ lưu trữ hỗ trợ lập chỉ mục không đặc biệt cho các cột không gian, công cụ này tạo ra một chỉ mục cây B. Chỉ mục cây B trên các giá trị không gian rất hữu ích cho việc tra cứu giá trị chính xác, nhưng không phải cho quét phạm vi.

Để xóa chỉ mục không gian, sử dụng [ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html)hoặc [DROP INDEX](https://dev.mysql.com/doc/refman/5.7/en/drop-index.html):

Với [ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html):

ALTER TABLE geom DROP INDEX g;

Với [DROP INDEX](https://dev.mysql.com/doc/refman/5.7/en/drop-index.html):

DROP INDEX g ON geom;

Ví dụ: Giả sử rằng một bảng geomchứa hơn 32.000 hình học, được lưu trữ trong cột gloại GEOMETRY. Bảng cũng có một AUTO\_INCREMENTcột fidđể lưu trữ các giá trị ID đối tượng.

mysql> DESCRIBE geom;

+-------+----------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------+----------+------+-----+---------+----------------+

| fid | int(11) | | PRI | NULL | auto\_increment |

| g | geometry | | | | |

+-------+----------+------+-----+---------+----------------+

2 rows in set (0.00 sec)

mysql> SELECT COUNT(\*) FROM geom;

+----------+

| count(\*) |

+----------+

| 32376 |

+----------+

1 row in set (0.00 sec)

Để thêm một chỉ mục không gian trên cột g, hãy sử dụng câu lệnh này:

mysql> ALTER TABLE geom ADD SPATIAL INDEX(g);

Query OK, 32376 rows affected (4.05 sec)

Records: 32376 Duplicates: 0 Warnings: 0

1. **Định dạng dữ liệu không gian được hỗ trợ**

Hai định dạng dữ liệu không gian tiêu chuẩn được sử dụng để thể hiện các đối tượng hình học trong các truy vấn:

* Định dạng văn bản nổi tiếng (WKT)
* Định dạng nhị phân nổi tiếng (WKB)

Trong nội bộ, MySQL lưu trữ các giá trị hình học theo định dạng không giống với định dạng WKT hoặc WKB. (Định dạng bên trong giống như WKB nhưng với 4 byte ban đầu để biểu thị SRID.)

Có các chức năng có sẵn để chuyển đổi giữa các định dạng dữ liệu khác nhau; thấy [Mục 12.16.6, “Geometry Chức năng chuyển đổi định dạng”](https://dev.mysql.com/doc/refman/5.7/en/gis-format-conversion-functions.html) .

Các phần sau đây mô tả các định dạng dữ liệu không gian mà MySQL sử dụng:

* [Định dạng văn bản nổi tiếng (WKT)](https://dev.mysql.com/doc/refman/5.7/en/gis-data-formats.html#gis-wkt-format)
* [Định dạng nhị phân (WKB) nổi tiếng](https://dev.mysql.com/doc/refman/5.7/en/gis-data-formats.html#gis-wkb-format)
* [Định dạng lưu trữ hình học bên trong](https://dev.mysql.com/doc/refman/5.7/en/gis-data-formats.html#gis-internal-format)

*Định dạng văn bản nổi tiếng (WKT)*

Biểu diễn các giá trị hình học được biết đến (WKT) được thiết kế để trao đổi dữ liệu hình học ở dạng ASCII. Đặc tả OpenGIS cung cấp một ngữ pháp Backus-Naur chỉ định các quy tắc sản xuất chính thức để viết các giá trị WKT (xem [Phần 11.5, Các kiểu dữ liệu không gian của Hồi giáo](https://dev.mysql.com/doc/refman/5.7/en/spatial-types.html) ).

Ví dụ về biểu diễn WKT của các đối tượng hình học:

* A Point:

POINT(15 20)

Các tọa độ điểm được chỉ định không có dấu phẩy phân tách. Điều này khác với cú pháp của [Point()](https://dev.mysql.com/doc/refman/5.7/en/gis-mysql-specific-functions.html#function_point)hàm SQL , yêu cầu một dấu phẩy giữa các tọa độ. Cẩn thận sử dụng cú pháp phù hợp với bối cảnh của một hoạt động không gian nhất định. Ví dụ, cả hai câu lệnh sau đều sử dụng [ST\_X()](https://dev.mysql.com/doc/refman/5.7/en/gis-point-property-functions.html#function_st-x)để trích xuất tọa độ X từ một Pointđối tượng. Việc đầu tiên tạo ra đối tượng trực tiếp bằng cách sử dụng [Point()](https://dev.mysql.com/doc/refman/5.7/en/gis-mysql-specific-functions.html#function_point)chức năng. Thứ hai sử dụng một đại diện WKT được chuyển đổi thành một Pointvới [ST\_GeomFromText()](https://dev.mysql.com/doc/refman/5.7/en/gis-wkt-functions.html#function_st-geomfromtext).

mysql> SELECT ST\_X(Point(15, 20));

+---------------------+

| ST\_X(POINT(15, 20)) |

+---------------------+

| 15 |

+---------------------+

mysql> SELECT ST\_X(ST\_GeomFromText('POINT(15 20)'));

+---------------------------------------+

| ST\_X(ST\_GeomFromText('POINT(15 20)')) |

+---------------------------------------+

| 15 |

+---------------------------------------+

* A LineStringcó bốn điểm:

LINESTRING(0 0, 10 10, 20 25, 50 60)

Các cặp tọa độ điểm được phân tách bằng dấu phẩy.

* A Polygonvới một vòng bên ngoài và một vòng bên trong:

POLYGON((0 0,10 0,10 10,0 10,0 0),(5 5,7 5,7 7,5 7, 5 5))

* A MultiPointvới ba Pointgiá trị:

MULTIPOINT(0 0, 20 20, 60 60)

Kể từ MySQL 5.7.9, các hàm không gian như [ST\_MPointFromText()](https://dev.mysql.com/doc/refman/5.7/en/gis-wkt-functions.html#function_st-mpointfromtext)và [ST\_GeomFromText()](https://dev.mysql.com/doc/refman/5.7/en/gis-wkt-functions.html#function_st-geomfromtext)chấp nhận các biểu diễn định dạng WKT của các MultiPointgiá trị cho phép các điểm riêng lẻ trong các giá trị được bao quanh bởi dấu ngoặc đơn. Ví dụ: cả hai lệnh gọi hàm sau đều hợp lệ, trong khi trước MySQL 5.7.9, lệnh thứ hai tạo ra lỗi:

ST\_MPointFromText('MULTIPOINT (1 1, 2 2, 3 3)')

ST\_MPointFromText('MULTIPOINT ((1 1), (2 2), (3 3))')

Kể từ MySQL 5.7.9, đầu ra cho MultiPointcác giá trị bao gồm dấu ngoặc đơn quanh mỗi điểm. Ví dụ:

mysql> SET @mp = 'MULTIPOINT(1 1, 2 2, 3 3)';

mysql> SELECT ST\_AsText(ST\_GeomFromText(@mp));

+---------------------------------+

| ST\_AsText(ST\_GeomFromText(@mp)) |

+---------------------------------+

| MULTIPOINT((1 1),(2 2),(3 3)) |

+---------------------------------+

Trước MySQL 5.7.9, đầu ra cho cùng một giá trị không bao gồm dấu ngoặc đơn quanh mỗi điểm:

mysql> SET @mp = 'MULTIPOINT(1 1, 2 2, 3 3)';

mysql> SELECT ST\_AsText(ST\_GeomFromText(@mp));

+---------------------------------+

| ST\_AsText(ST\_GeomFromText(@mp)) |

+---------------------------------+

| MULTIPOINT(1 1,2 2,3 3) |

+---------------------------------+

* A MultiLineStringcó hai LineStringgiá trị:

MULTILINESTRING((10 10, 20 20), (15 15, 30 15))

* A MultiPolygoncó hai Polygongiá trị:

MULTIPOLYGON(((0 0,10 0,10 10,0 10,0 0)),((5 5,7 5,7 7,5 7, 5 5)))

* A GeometryCollectionbao gồm hai Pointgiá trị và một LineString:

GEOMETRYCOLLECTION(POINT(10 10), POINT(30 30), LINESTRING(15 15, 20 20))

*Định dạng nhị phân (WKB) nổi tiếng*

Biểu diễn các giá trị hình học nhị phân (WKB) nổi tiếng được sử dụng để trao đổi dữ liệu hình học dưới dạng luồng nhị phân được biểu thị bằng [BLOB](https://dev.mysql.com/doc/refman/5.7/en/blob.html)các giá trị chứa thông tin WKB hình học. Định dạng này được xác định bởi đặc tả OpenGIS.

WKB sử dụng số nguyên không dấu 1 byte, số nguyên không dấu 4 byte và số chính xác kép 8 byte (định dạng IEEE 754). Một byte là tám bit.

Ví dụ: giá trị WKB tương ứng POINT(1 -1)bao gồm chuỗi 21 byte này, mỗi byte được biểu thị bằng hai chữ số thập lục phân:

0101000000000000000000F03F000000000000F0BF

Trình tự bao gồm các thành phần được hiển thị trong bảng sau.

Bảng 11.2 Ví dụ về các thành phần WKB

| Thành phần | Kích thước | Giá trị |
| --- | --- | --- |
| Đặt hàng byte | 1 byte | 01 |
| Loại WKB | 4 byte | 01000000 |
| Tọa độ X | 8 byte | 000000000000F03F |
| Tọa độ Y | 8 byte | 000000000000F0BF |

Đại diện thành phần như sau:

Chỉ báo thứ tự byte là 1 hoặc 0 để biểu thị lưu trữ cuối nhỏ hoặc cuối lớn. Các đơn đặt hàng byte cuối nhỏ và lớn cuối còn được gọi là Biểu diễn dữ liệu mạng (NDR) và Biểu diễn dữ liệu ngoài (XDR), tương ứng.

Loại WKB là một mã chỉ ra loại hình học. MySQL sử dụng các giá trị từ 1 đến 7 để chỉ Point, LineString, Polygon, MultiPoint,MultiLineString, MultiPolygon, và GeometryCollection.

Một Pointgiá trị có tọa độ X và Y, mỗi giá trị được biểu thị dưới dạng giá trị chính xác kép.

Các giá trị WKB cho các giá trị hình học phức tạp hơn có cấu trúc dữ liệu phức tạp hơn, như được nêu chi tiết trong đặc tả OpenGIS.

Định dạng lưu trữ hình học bên trong

MySQL lưu trữ các giá trị hình học bằng cách sử dụng 4 byte để chỉ ra SRID theo sau là biểu diễn WKB của giá trị. Để biết mô tả về định dạng WKB, hãy xem [Định dạng nhị phân được biết đến (WKB)](https://dev.mysql.com/doc/refman/5.7/en/gis-data-formats.html#gis-wkb-format) .

Đối với phần WKB, những cân nhắc dành riêng cho MySQL này được áp dụng:

* Byte chỉ báo thứ tự byte là 1 vì MySQL lưu trữ hình học dưới dạng các giá trị nhỏ.
* MySQL hỗ trợ các loại hình học của Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, vàGeometryCollection. Các loại hình học khác không được hỗ trợ.

Các [LENGTH()](https://dev.mysql.com/doc/refman/5.7/en/string-functions.html#function_length)chức năng trả về không gian trong byte cần thiết để lưu trữ giá trị. Thí dụ:

mysql> SET @g = ST\_GeomFromText('POINT(1 -1)');

mysql> SELECT LENGTH(@g);

+------------+

| LENGTH(@g) |

+------------+

| 25 |

+------------+

mysql> SELECT HEX(@g);

+----------------------------------------------------+

| HEX(@g) |

+----------------------------------------------------+

| 000000000101000000000000000000F03F000000000000F0BF |

+----------------------------------------------------+

Độ dài giá trị là 25 byte, được tạo thành từ các thành phần này (có thể được nhìn thấy từ giá trị thập lục phân):

* 4 byte cho số nguyên SRID (0)
* 1 byte cho thứ tự byte số nguyên (1 = little endian)
* 4 byte cho thông tin loại số nguyên (1 = Point)
* 8 byte cho tọa độ X chính xác kép (1)
* 8 byte cho tọa độ Y có độ chính xác kép (1)

1. **Danh mục tài liệu tham khảo**

<https://vietjack.com/mysql/chi_muc_index_trong_mysql.jsp>

<https://dev.mysql.com/doc/refman/5.7/en/using-spatial-indexes.html>