

13 - Indexing for High Performance

Thursday, April 18, 2024 10:04 AM

What is Indexes ?

Indexes are a powerful tool used in the background of a database to speed up querying. Indexes power queries by providing a method to quickly lookup the requested data.

Simply put, an index is a pointer to data in a table. An index in a database is very similar to an index in the back of a book.

How To Create Them ?

```
CREATE INDEX <index_name>  
ON <table_name> (column1, column2, ...)
```

```
CREATE INDEX idx_state  
ON customers (state) ;
```

Show indexes & Analyzing Tables

How To Show Indexes on Certain Table ?

```
SHOW INDEXES IN < Table_name>;  
SHOW INDEXES IN orders;
```

How To analyze table with accurate statistics ?

```
ANALYZE TABLE <table_name>  
ANALYZE TABLE orders ;
```

Index on Strings Columns

- **Prefix Index**

The Problem is that index on string is so cost for memory and time but with simple trick you can do better
You can just make the index on the prefix of the string not the full string and you can apply that for

[CHAR , VARCHAR , TEXT , BLOB]

2

```
3 • CREATE INDEX prefix_idx ON customers(last_name(20)) ;
```

4

5

FULL TEXT SEARCH

Full-text search is a searching technique used to examine all words in the document that might not perfectly match the

search criteria. The records contain textual data like product descriptions, blog posts, articles, etc. MySQL provides support for full-text indexing and searching from version 3.23.23. In this section, we are going to learn the introduction of FULLTEXT search, syntax, advantages, disadvantages, features, and its types.

[[SEAECH MORE ABOUT THIS TYPE OF INDEX](#)]

```
CEARTE FULLTEXT INDEX idx_col1_col2 ON <table_name> (col1,col2);
```

```
SELECT *  
FROM posts  
WHERE MATCH(col1,col2) AGAINST ('searched words')
```

[FULL-TEXT SEARCH](#)
[FULL-TEXT SEARCH](#)

Composite Index

The idea is that assume you query based on condition of two columns and you have index for each column of them Index of col1 is good but you can do better and also index of col2 is also good but you can do better If you combine them to be one index you become much faster and that the idea of the composite index

```
CREATE INDEX IDX_STATE_POINTS ON customer (state , points) ;
```

Is Order Of The Column Matter ?

Yes it's Matter and affect a lot at the speed of the query



[!] NOTE

You Mustn't Use This Rules Blindly you must analyze how MYSQL will execute your query and them
Come up with best order for your case [One Index Not Best For all cases as system grow you will need more indexes]