Index Cardinality, Table Fragmentation, Use INDEX Hint, ForceIndex Hint

Index Cardinality

- Index cardinality refers to the uniqueness of values stored in a specified column within an index.
- MySQL generates the index cardinality based on statistics stored as integers, therefore, the value may not be necessarily exact.
- The query optimizer uses the index cardinality to generate an optimal query plan for a given query
- It also uses the index cardinality to decide whether to use the index or not in the join operations.
- If the query optimizer chooses the index with a low cardinality, it is may be more effective than scan rows without using the index.
- To view the index cardinality, you use the SHOW INDEXES command.

Table Fragmentaion

- Whenever MySQL deletes rows from your table, the space left behind is then empty.
- Over time with a lot of DELETEs, this space can grow larger than the used space in your table.
- When MySQL goes to scan that data, it scans to the high water mark of the table, that is the highest point at which data has been added.
- If new inserts occur, MySQL will try to use that space, but nevertheless gaps will persist.
- This extra fragmented space can make reads against the table less efficient than they might otherwise be.
- select table_schema, table_name, data_free, engine

  ```
          from information_schema.tables where table_schema
          not in ('information_schema', 'mysql') and data_free > 0;
  ```
- ALTER TABLE tbl_name ENGINE=INNODB
- optimize table tablename;

USE INDEX Hint

- In MySQL, when you submit an SQL query, the query optimizer will try to make an optimal query execution plan.
- To determine the best possible plan, the query optimizer makes use of many parameters.
- One of the most important parameters for choosing which index to use is stored key distribution which is also known as cardinality.
- The cardinality, however, may be not accurate for example in case the table has been modified heavily with many inserts or deletes.
- To solve this issue, you should run the ANALYZE TABLE statement periodically to update the cardinality.
- In addition, MySQL provides an alternative way that allows you to recommend the indexes that the query optimizer should by using an index hint called USE INDEX.
- SELECT select_list

```
        FROM table_name USE INDEX(index_list)

        WHERE condition;
```

FORCE INDEX Hint

- The query optimizer is a component in the MySQL Database server that makes the most optimal execution plan for an SQL statement.
- The query optimizer uses the available statistics to come up with the plan that has the lowest cost among all candidate plans.
- For example, a query might request for products whose prices are between 10 and 80.
- If the statistics show that 80% of products have these price ranges, then it may decide that a full table scan is the most efficient.
- However, if statistics show that very few products have these price ranges, then reading an index followed by a table access could be faster and more efficient than a full table scan.
- In case the query optimizer ignores the index, you can use the FORCE INDEX hint to instruct it to use the index instead.
- SELECT * FROM table_name FORCE INDEX (index_list) WHERE condition;