阿里一面：MySQL 一张表最多支持多少个索引？16个？64个？还是无限制？

原创 Fox Fox爱分享 2025年05月04日 07:01 湖南

Fox爱分享
分享微服务、中间件、消息队列、搜索引擎、分布式存储和高并发架构，云原生，AI大模...
209篇原创内容

公众号

在MySQL数据库设计中，合理使用索引是提升查询性能的关键手段之一。然而，许多开发者在设计表结构时，可能会对MySQL表支持的索引数量存在疑问。本文将详细解析MySQL中不同存储引擎和版本对索引数量的限制，并提供一些实际应用中的建议。

InnoDB存储引擎

1. 二级索引限制

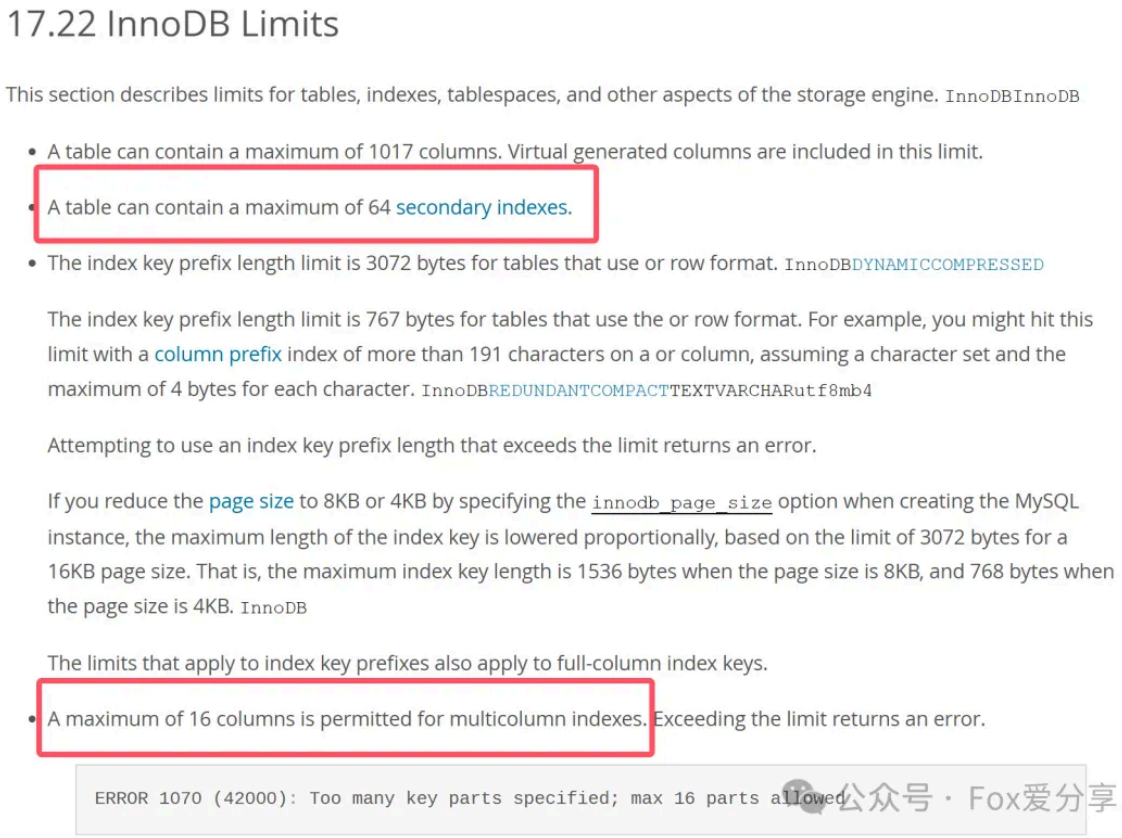
InnoDB存储引擎支持最多 **64个二级索引**（即非主键索引）。主键索引（聚集索引）不计入此限制。

2. 总索引数

如果表有主键，则总索引数为 **65个**（64个二级索引 + 1个主键索引）。

3. 复合索引列数限制

单个索引最多包含 **16列**。如果超过16列，MySQL会报错。



17.22 InnoDB Limits

This section describes limits for tables, indexes, tablespaces, and other aspects of the storage engine. InnoDBInnoDB

- A table can contain a maximum of 1017 columns. Virtual generated columns are included in this limit.
- A table can contain a maximum of 64 secondary indexes.
- The index key prefix length limit is 3072 bytes for tables that use or row format. InnoDBDYNAMICCOMPRESSED

  The index key prefix length limit is 767 bytes for tables that use the or row format. For example, you might hit this limit with a column prefix index of more than 191 characters on a or column, assuming a character set and the maximum of 4 bytes for each character. InnoDBREDUNDANTCOMPACTTEXTVARCHARutf8mb4

  Attempting to use an index key prefix length that exceeds the limit returns an error.

  If you reduce the page size to 8KB or 4KB by specifying the innodb_page_size option when creating the MySQL instance, the maximum length of the index key is lowered proportionally, based on the limit of 3072 bytes for a 16KB page size. That is, the maximum index key length is 1536 bytes when the page size is 8KB, and 768 bytes when the page size is 4KB. InnoDB

  The limits that apply to index key prefixes also apply to full-column index keys.
- A maximum of 16 columns is permitted for multicolumn indexes. Exceeding the limit returns an error.

```
ERROR 1070 (42000): Too many key parts specified; max 16 parts allowed
```


innoDB限制的官方文档:

https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html

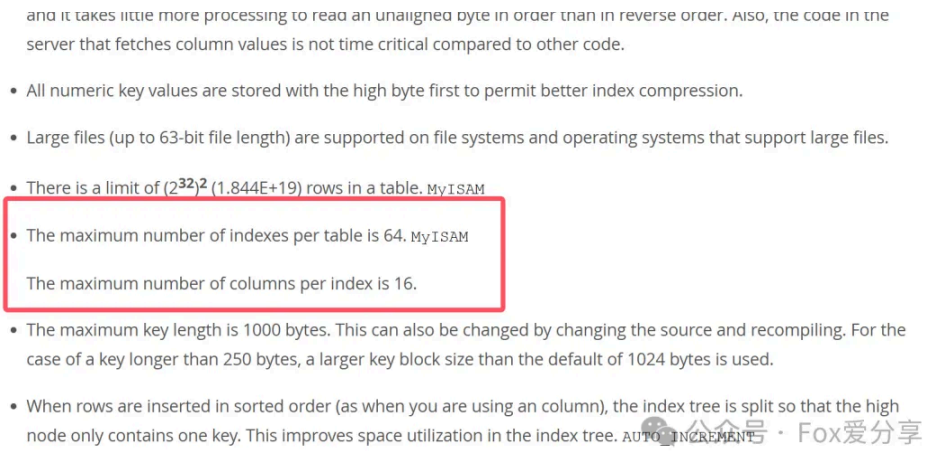
MyISAM存储引擎

1. 总索引数限制

MyISAM存储引擎支持最多 **64个索引**（包括主键索引）。

2. 复合索引列数限制

单个索引最多包含 **16列**。



and it takes little more processing to read an unaligned byte in order than in reverse order. Also, the code in the server that fetches column values is not time critical compared to other code.

- All numeric key values are stored with the high byte first to permit better index compression.
- Large files (up to 63-bit file length) are supported on file systems and operating systems that support large files.
- There is a limit of $(2^{32})^2$ (1.844E+19) rows in a table. MyISAM
- The maximum number of indexes per table is 64. MyISAM

  The maximum number of columns per index is 16.
- The maximum key length is 1000 bytes. This can also be changed by changing the source and recompiling. For the case of a key longer than 250 bytes, a larger key block size than the default of 1024 bytes is used.
- When rows are inserted in sorted order (as when you are using an column), the index tree is split so that the high node only contains one key. This improves space utilization in the index tree. AUTO_INCREMENT

myisam限制的官方文档:

https://dev.mysql.com/doc/refman/8.0/en/myisam-storage-engine.html

版本差异

1. MySQL 5.7及之前版本

- **InnoDB** 和 **MyISAM** 均支持最多 **64个索引**（InnoDB的主键索引不计入此限制）。

2. MySQL 8.0及之后版本

- **InnoDB** 默认启用 **innodb_large_prefix** 特性，允许更长的索引键长度，但索引数量限制仍为 **64个二级索引**。
- **MyISAM** 的索引数量限制保持不变。

常见误解澄清

16个索引的误区

部分资料提到“每个表最多16个索引”，这可能是混淆了索引数量与复合索引的列数限制。实际上，**单个索引最多包含16列**，而不是表最多只能有16个索引。

实际应用建议

虽然MySQL理论上允许较多的索引，但在实际应用中，建议单表索引数量不超过 **5–6个**，以避免以下问题：

1. **写性能下降**：插入、更新和删除操作时，MySQL需要同步更新所有相关索引，过多的索引会显著增加这些操作的开销。
2. **存储开销**：每个索引都会占用额外的磁盘空间和内存资源，过多的索引可能导致磁盘空间不足或内存压力过大。
3. **查询优化器选择错误**：过多的索引可能导致查询优化器选择错误的索引，反而降低查询性能。

总结

- **InnoDB**：最多 **64个二级索引 + 1个主键索引 = 65个索引**。
- **MyISAM**：最多 **64个索引（含主键）**。
- **复合索引**：单个索引最多 **16列**。

在设计表结构时，应根据具体的业务需求合理设计索引，避免过度使用。过多的索引不仅会增加维护成本，还可能对性能产生负面影响。建议通过分析查询语句，确定哪些字段经常用于WHERE条件、JOIN操作或ORDER BY排序，并为这些字段创建索引。同时，定期检查表中的索引，删除那些冗余或很少使用的索引，以优化索引策略。

如果你觉得这篇文章对你有所帮助，欢迎点个“推荐”或分享给更多的小伙伴！关注公众号「**Fox爱分享**」，解锁更多干货，一起在技术的道路上不断前行！

Mysql · 目录

上一篇
阿里一面：千万级大表如何快速删除大量数据

下一篇
UUIDv7：打破谣言，我就是要用UUID做主键！凭什么它是最强主键候选者？