



链滴

MYSQL 索引原理及建索引原则剖析

作者: [dreamertn9527](#)

原文链接: <https://ld246.com/article/1513079718486>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

MYSQL索引原理及建索引原则剖析

索引类型

- 非聚集索引：像字典一样最前面的几页索引，或者书的目录
- 聚集索引：像书的页码下面的数字

Innodb表存储引擎（聚集索引）

索引保存的类型：

1. 索引列的值
2. 指向数据行的指针

Innodb中只有一个聚集索引：

1. 默认会拿主键ID作为聚集索引
2. 如果没有主键，会取非空的唯一索引作为聚集索引
3. 如果上面的都没有，innodb会自己维护一个唯一ID来作为聚集索引

当存在多个索引时：

因为Innodb只有一个聚集索引，所以聚集索引会包含索引列，以及指向行数据的指针，而其他的索引仅仅含索引列，对应的其他数据则是指向相应的聚集索引，通过聚集索引查找对应的其他值

Innodb索引特点：

1. idb文件：既存索引，又存对应的数据

Mylsam的索引特点：

1. MYI: 存储索引
2. MYD: 存储数据
3. FRM: 存储数据结构

索引的优缺点及建索引的原则

索引的优点：提高检索速度，降低磁盘I/O

索引的缺点：索引需要存储，需要磁盘空间，索引实际上就是一张表，当字段更新(INSERT, DELETE, PDATE)的时候，存在性能损耗

建索引的原则：

适合建立索引：

1. 频繁作为WHERE条件的字段
2. 关联字段可以建索引，例如外键

3. ORDER BY , GROUP BY(底层实现是先做排序, 然后分组) 后面的可以建索引

不适合建立索引:

1. WHERE条件中用不到的字段不适合建立索引
2. 频繁更新的字段不适合建立索引
3. 数据值分布比较均匀的不适合建立索引(区分度不高)
4. 表的数据可以确定行数的, 而且数据量很少

索引失效

什么情况下索引失效

1. Order by

```
SELECT * FROM USER ORDER BY NAME;
```

上面的情况索引生效, 如果存在下面的索引:

如果加上Index(name, age)复合索引

则下面的查询索引失效

```
SELECT * FROM USER ORDER BY AGE; -- 失效  
SELECT * FROM USER WHERE AGE = 18 AND NAME = 'ZHANGSANG' -- 失效
```

必须按照下面的排序, 则不会索引失效

```
SELECT * FROM USER ORDER BY NAME, AGE;
```

为什么会失效呢, 比如这个复合索引像一个楼梯。第一层使用NAME, 第二层使用AGE, 则必须先上层, 才能上二层

替换索引为index(age, name),sql改为如下方式:

```
SELECT * FROM USER WHERE AGE > 18 AND NAME = 'ZHANGSANG' -- 前面的范围索引有效  
后半部分索引失效
```