# 索引

## 1.1索引的作用

索引是找出一列中包含某个特定值的行。

## 1.2索引的优缺点

优点：

1. 可以对数据库中表的任意字段建立索引。
2. 大大加快查询速度。

缺点：

1. 索引的创建和维护需要时间。
2. 索引会占用空间，如果索引太多，会达到空间上线。
3. 如果表中的数据更新，索引也要更新，会耗费时间。

## 1.3使用原则

1. 不要再数据更新频繁的表使用，因为那样会大大增加索引的维护开销。

2. 不要在不同值少的列中使用，例如性别，只有男，女两种不同值，要在不用值多的列种使用。

3. 数据量小的表不要使用。因为可能遍历整个表的时间会比遍历索引的时间短。

## 1.4索引的分类

Innodb和MyISAM都是使用的bTree索引。

Heap和Memory支持BTree索引和Hash索引。

#### 1.4.1单列索引（包括普通，主键和唯一三种索引）

一个索引只包含单个列。

## 1.4.2组合索引（包括普通，主键和唯一三种索引）

在多个字段组合上建立的索引，只有在查询条件中包含组合的左边字段上，才会使用索引。

## 1.5索引操作（创建和删除）

1.5.1创建索引

创建表的时候创建索引

格式：CREATE TABLE 表名[字段名 数据类型] [UNIQUE|FULLTEXT|SPATIAL|...] [INDEX|KEY] [索引名字] (字段名[length]) [ASC|DESC]

添加索引

-在创建表后在创建

create index name on s1(name); #添加普通索引

create unique age on s1(age);添加唯一索引

alter table s1 add primary key(id); #添加住建索引，也就是给id字段增加一个主键约束

create index name on s1(id,name); #添加普通联合索引

1.5.2删除索引

drop index id on s1;

drop index name on s1; #删除普通索引

drop index age on s1; #删除唯一索引，就和普通索引一样，不用在index前加unique来删，直接就可以删了

alter table s1 drop primary key; #删除主键(因为它添加的时候是按照alter来增加的，那么我们也用alter来删)

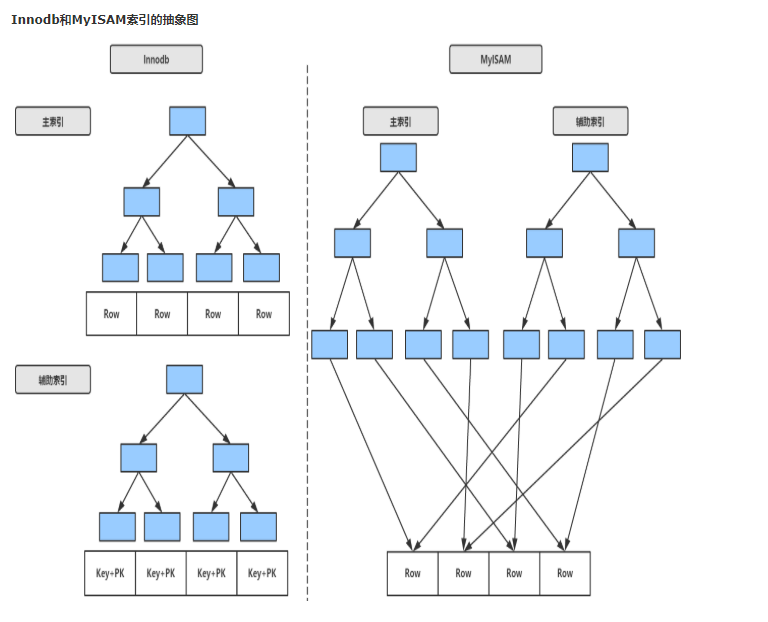
## 1.6索引的存储结构和原理

在innodb和myisam引擎中存储的都是用B+树来存储的。

Innodb的叶子结点存的是索引和行的内容。

MYISAM的叶子结点存的是索引和行的地址。

这两个存储殷勤的表的原始数据文件的也是b+树的方式存的。



# 2.乐观锁、悲观锁、共享锁、排他锁、行锁、表锁

## 乐观锁

什么是乐观锁？

乐观的认为当前更新操作不会发生冲突，在更新后再去检查是否发生冲突。

如何实现乐观锁

给数据库加一个version字段，每更新一次，就把version字段加1。也就是先查询出那一条记录，获取出version字段，如果要对那条记录进行更新，则先判断此刻的version值与前面的version值是否相等。若相等则执行更新操作，并把version+1。若不相等，则不执行更新操作。

## 悲观锁

悲观锁就是认为此次数据库操作必然会发生冲突，所以每次数据库操作之前都要获取锁。悲观锁是数据库自己实现了的。可直接调用。

共享锁和排它锁就是悲观锁的两种不同的实现。

#### 共享锁

共享锁就是不同的资源针对同一个资源共享同一把锁。

共享锁只需要你在执行的语句后面加上lock in share mode就可以了。

#### 排它锁

不同的事务对同一个资源只需要有一把锁就可以了。

使用时，只需要在执行的语句后面加一个for update就可以了。

# 3.存储殷勤Innodb和MYISAM的差异、

区别：

1. Innodb支持事务，而MYISAM不支持，事务出错了还可以回滚。
2. MyISAM适合查询为主的应用，而Innodb适合频繁修改以及涉及安全性较高的应用。
3. Innodb支持外健，而MyISAM不支持。
4. 从MySql5.5.5以后，Innodb是弄人殷勤。
5. Innodb不保存表的行数，而MyISAM不保存。
6. 清空整个表时，Innodb是一行一行的删除，而MyISAM则会重建表。
7. Innodb支持行锁和表锁。MyISAM只支持表锁。