## 数据库引擎

概念：数据库引擎就是数据库中表存放数据的一种格式，不同的存放格式，对应的特性不同。比如默认的引擎InnoDB存储引擎，支持事务，外键，行级锁。

存储引擎是mysql数据库的组件之一，是组件就意味着，可以更换，拔插。对应正在运行的数据库可以加载或者卸载所需要的引擎文件。

种类：察看数据库支持的引擎 show engines；

InnoDB：默认引擎，支持事务（提交，回滚），行级锁，外键约束。功能比较全，但是带来的负面效应就是效率一般。因为支持事务，具有回滚，提交功能。提高了对并发的处理能力，维护了数据的完整性。非常适合业务逻辑强，修改操作频繁的项目。

存储格式：.frm 用来存储表的结构 所有的表，共用这一个表结构存储文件。

增加表空间的大小：可以设置为自动增加

多版本控制，有点类似于git的版本控制，会创建三个指针，来完成对事务回滚的支持。

MyISAM: Mysql5.5前的版本中的默认存储引擎为MyISAM，之后就采用InnoDB作为默认存储引擎。MyISAM不支持事务的完整性和并发性。但是处理的速度更快。适合论坛，博客等去写操作非常多的项目

.myd 存储数据 mydate的缩写

.myi 存储索引 myindex的缩写

## 数据库中的索引

### 分类

索引：类似于我们的目录，可以快速定位内容

将记录中的字段和这条字段的存储位置建立联系，并按照一定顺序排列好。

实际上就是存储在引擎中的数据结构

所以分为btree索引，大部分引擎都支持btree索引

hash索引，只有memory和ndb引擎支持，使用的场景简单

fulltext索引：全文索引 myisam支持全文索引，mysql从5.6之后也支持全文索引

rtree索引：空间索引

索引的分类：普通索引，唯一索引，主键索引，全文索引，空间索引

根据字段分：单列索引和复合索引。

从最左边开始截取字段：最左索引

什么是覆盖索引：就是我们的索引，能够包含所有要查询的数据，这个索引就叫做覆盖索引

### 索引的建立

当我们建立主键约束，外键约束，唯一约束的时候，会自动建立索引。

主键索引和普通索引

Primary key [ 索引类型 ] （字段列表） [ 索引选项 ]

[ key | index ] [ 索引名称 ] [ 索引类型 ] （字段选项）[ 索引选项 ]

Primary key btree id comment ‘主键索引’

Index name\_index (name)

主键索引只能有一个：因为只用一个主键

唯一索引可以有多个，但是这多个唯一索引对应的字段不能重复。

普通索引

全文索引：类似于like的模糊查询，目前只能支持英文，不能支持中文

建立：add fulltext index [ 索引名称 ] （字段列表）[ 索引选项 ]

使用全文索引： match （字段列表）against （字符串）

其中字符串只能是个完整的单词或者句子，不能是部分单词。

空间索引：要建立在空间字段上面。空间字段类型：geometry

复合索引：最左前缀原则，就是把最频繁使用的字段放在最左侧，只有在第一个字段使用的时候，给索引才被使用。

前缀索引：就是将一个字段的一部分作为关键字，当作索引。因为有些字段，很长，这样导致很浪费空间，同时查询也很慢。那么就截取字段的一部分，把这一部分当作索引。

Add index (username(3)) comment’截断前三位’ **全文索引和空间索引不支持前缀索引**

怎么计算截断前几位呢？ count（distinct username）/ count (username)

Distinct left(username , 1) / count(username)

### 索引的察看

Show { index | indexs | keys } from 表名

Non\_unique: 0表示false 唯一不能重复。1表示true，能重复

Key\_name: 索引的名字，如果是主键索引，名字为primary

Colletion：表示字段是否排序，其中A表示排序，null表示不排序

Sub\_part：前缀索引的截取的字段的长度；若字段都被引用，则为null

Index\_type: 索引的类型

Coment：索引的注释信息

Index\_comment: 添加索引时候我们添加的信息

## 删除索引

因为索引很占用系统资源，所以对于不常用的索引，要及时删除

**删除主键索引和删除非主键索引**

删除主键索引：先删除主键索引的属性：auto\_increment; 再将主键索引删除

Alter talbe 表名 modify 字段名 字段类型

Alter table 表名 drop primary key或者 drop index `primary` on 表名

## 索引的使用原则

索引能加快查询的速度，但是相应的也会占用系统资源，同时给数据的危害带来不方便。

数字型字段，存储空间较小的字段，查询条件中使用频繁的字段适合使用索引

重复度高的字段，跟新频繁的字段不适合建立索引

索引要建立在独立的字段中，字段不能是表达式的一部分或者是函数的参数

## 数据库中的锁

数据库中的数据对于所用用户是共享的，当多个用户在同一个时间对同一个数据进行操作的时候，就会相互影响。就是并发。锁就是通过规定优先级，来解决并发过程的问题

共享锁：读锁；加了此锁，只能对数据进行读取操作，不能对数据进行修改（否则会报错），同时也不能读取其他表。其他客户端也可以读取该表的数据，其他客户端对该表进行修改操作时候，修改操作会被挂起，其他客户端可以进行对数据的插入操作。

排他锁：写锁。客户端能对数据进行增删改查。其他客户端的增删改查都会被挂起。等锁释放后才能执行。