# MySql数据库的优化技术

## 对mysql优化综合性的技术，主要包括：

1. 表的设计合理化（符合3NF）
2. 添加适当索引（index）【四种：普通索引、主键索引、唯一索引、unique、全文索引、空间索引】
3. 分表技术（水平分割，垂直分割）
4. 读写【写：update、delete、insert】分离
5. 存储过程【模块化编程、执行编译好的sql。 但是移植性不好】
6. 对mysql配置优化【my.ini：配置最大并发数，推荐1000左右，调整缓存大小】
7. Mysql服务器硬件升级
8. 定时清除不需要的数据，定时进行碎片整理。

## 表的设计合理化（符合3NF）

**分类：**

1. 第一范式（1NF）：每一列都是不可分割的原子数据项

2. 第二范式（2NF）：在1NF的基础上，非码属性必须完全依赖于码（在1NF基础上消除非主属性对主码的部分函数依赖）

\* 几个概念：

1. 函数依赖：A-->B,如果通过A属性(属性组)的值，可以确定唯一B属性的值。则称B依赖于A

例如：学号-->姓名。 （学号，课程名称） --> 分数

2. 完全函数依赖：A-->B， 如果A是一个属性组，则B属性值得确定需要依赖于A属性组中所有的属性值。

例如：（学号，课程名称） --> 分数

3. 部分函数依赖：A-->B， 如果A是一个属性组，则B属性值得确定只需要依赖于A属性组中某一些值即可。

例如：（学号，课程名称） -- > 姓名

4. 传递函数依赖：A-->B, B -- >C . 如果通过A属性(属性组)的值，可以确定唯一B属性的值，在通过B属性（属性组）的值可以确定唯一C属性的值，则称 C 传递函数依赖于A

例如：学号-->系名，系名-->系主任

5. 码：如果在一张表中，一个属性或属性组，被其他所有属性所完全依赖，则称这个属性(属性组)为该表的码

例如：该表中码为：（学号，课程名称）

\* 主属性：码属性组中的所有属性

\* 非主属性：除过码属性组的属性

3. 第三范式（3NF）：在2NF基础上，任何非主属性不依赖于其它非主属性（在2NF基础上消除传递依赖）

### Sql语句本身的优化

问题：如何从一个大项目中迅速定位执行速度慢的语句（定位慢查询）

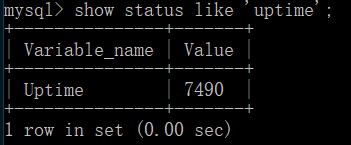
1. 首先我们了解mysql数据库的一些运行状态如何查询（比如想知道当前mysql运行的时间，一共执行了多少次select、update、delete、当前连接）

SHOW STATUS;

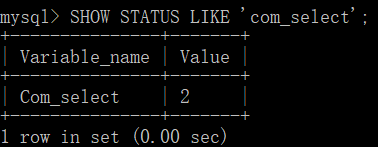
SHOW [SESSION|GLOBAL] LIKE ... -- 默认session

常用的：

* + - SHOW STATUS LIKE 'uptime'; -- 查询时间（mysql启动时间）



* + - SHOW STATUS LIKE 'com\_select'; -- 查看执行了多少次查询

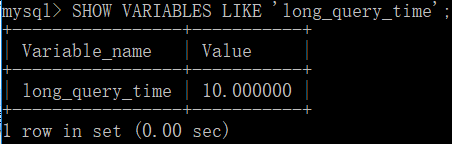


* + - SHOW STATUS LIKE 'com\_update'; -- 查看执行了多少次更新
    - SHOW STATUS LIKE 'com\_insert'; -- 查看执行了多少次插入
    - SHOW STATUS LIKE 'com\_delete'; -- 查看执行了多少次删除
    - SHOW STATUS LIKE 'connections'; -- 查看执行连接次数
    - SHOW STATUS LIKE 'slow\_queries'; -- 显示慢查询次数

### 如何去定位慢查询

构建一个大表（400万）🡪存储过程构建

mysql默认10秒为慢查询：SHOW VARIABLES LIKE 'long\_query\_time'; -- 查看慢查询时间



* 修改mysql的慢查询时间

SET long\_query\_time = 1; -- 设置慢查询时间

1. 构建一个大表测试（查看sql文件）

Mysql默认不会记录慢查询，需要在mysql启动时设置

Mysql5.5 可以在my.ini指定：bin\

mysqld.exe --safe-mode --slow-query-log



低版本msql5.0可以在my.ini指定：bin\mysqld.exe –log-slow-queries=d:/abc.log

* 先关闭mysql，再启动： bin\mysqld.exe –safe-mode –slow-query-log
* 日志存储路径：my.ini文件中

#Path to the database root

datadir="C:/ProgramData/MySQL/MySQL Server 5.5/Data/"

## 添加适当索引（index）【四种：普通索引、主键索引、唯一索引（unique）、全文索引、（其他：空间索引）】

1. **添加**

**主键索引**：当一张表，把某个列设为主键的时候，则该列就是主键索引

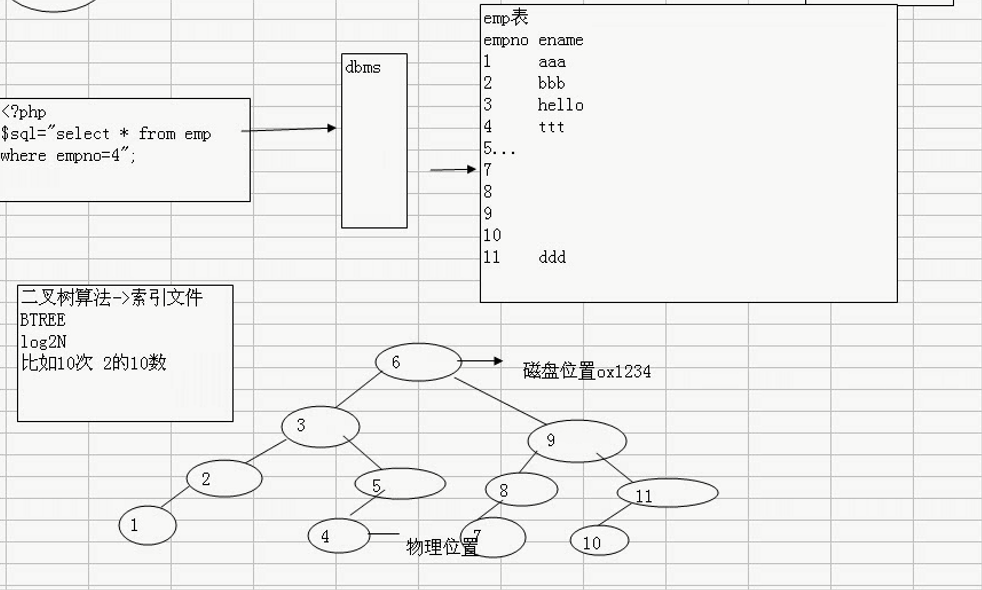
**主键所以不能为null切不能重复。**

-- 建表是创建主键

CREATE TABLE aaa(

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

NAME VARCHAR(32) NOT NULL DEFAULT '');



### 普通索引：

CREATE INDEX 索引名 ON 表名(列名1，列名2…);

### 唯一索引：

当某一列被指定问uniqre约束是，这列就是唯一索引，**值可以为null(一条或多条)，也可以为’’(空字符串)，空字符串只能插入一条记录**；

唯一索引可以在建表的的时候指定唯一约束，也可以在已建好的表上新建：

Create unique index 索引名 on 表名(列表)；

### 全文索引：

全文索引主要针对文本的检索，全文索引只对myisam有用

-- 创建全文索引表

CREATE TABLE articles (

id INT UNSIGNED AUTO\_INCREMENT NOT NULL PRIMARY KEY,

title VARCHAR(200),

body TEXT,

FULLTEXT (title,body)

) ENGINE=MYISAM;

-- 初始化数据

INSERT INTO articles (title,body) VALUES

('MySQL Tutorial','DBMS stands for DataBase ...'),

('How To Use MySQL Well','After you went through a ...'),

('Optimizing MySQL','In this tutorial we will show ...'),

('1001 MySQL Tricks','1. Never run mysqld as root. 2. ...'),

('MySQL vs. YourSQL','In the following database comparison ...'),

('MySQL Security','When configured properly, MySQL ...');

-- 全文索引使用

-- 错误用法：

SELECT \* FROM articles WHERE body LIKE '%database%';

-- 证明

EXPLAIN SELECT \* FROM articles WHERE body LIKE '%database%';

-- 正确用法

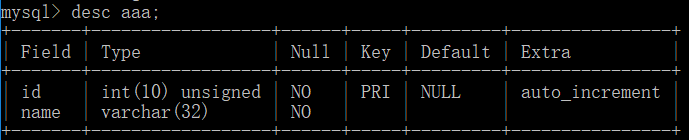
EXPLAIN SELECT \* FROM articles WHERE MATCH(title,body) AGAINST('database');

**注意：**

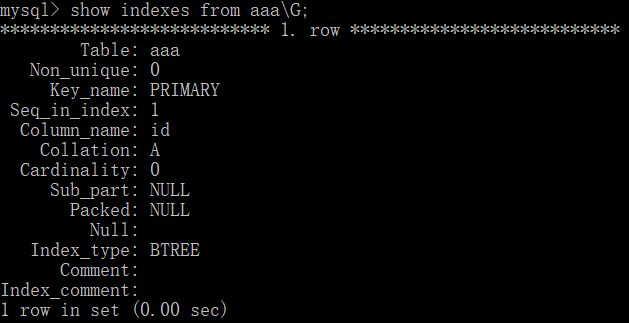
* 在mysql中fulltext索引只针对myisam生效
* Mysql自己提供的fulltext针对英文生效-->sphinx(coreseek)技术处理中
* 使用方法是match(字段名…)against(‘关键字’)
* 全文索引一个叫“停止词”，因为在一个文本中，创建索引是一个无穷大的数，因此，对一些常用词和字符，就不会创建这些词，称为停止词。

1. **查询索引**

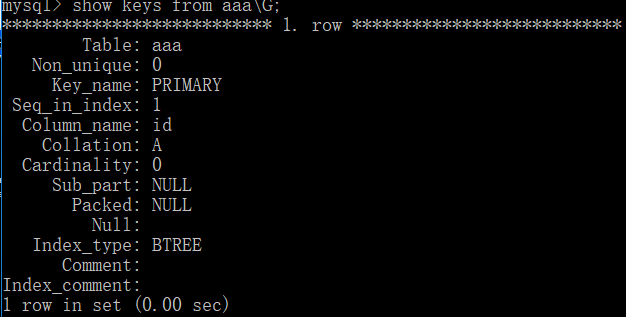
* DESC 表名;【缺点：不能显示索引名】



* SHOW INDEX(ES) FROM 表名\G;



* SHOW KEYS FROM表名;



1. **删除**

Alter table 表名 drop index 索引名；

如果删除主键索引

Alter table 表名 drop primary key

1. **修改**

先删除，后创建。

### 索引创建注意事项：

创建索引的代价：

1. 占用磁盘空间
2. 对DML操作有影响（变慢）

哪些列上适合添加索引？

1. 当做where查询条件经常使用的字段应该创建索引
2. 唯一性太差的字段不适合单独创建索引，即使频繁作为查询条件
3. 更新非常频繁的字段不适合创建索引
4. Where字句中不出现的字段不创建索引

### 索引使用注意事项：

下列几种情况有可能使用到索引：

1. 对于创建的多列索引，只要查询条件使用了最左的列，索引一般就会被使用

Alter table dept add index my\_ind(dname,loc); //dname左边的列，loc右边的列

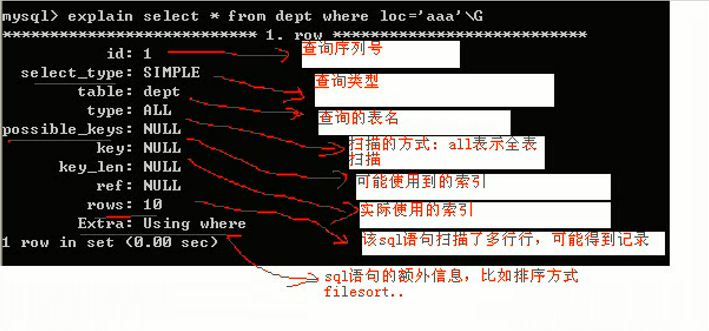
1. 对于like查询，如果是’%aaa’不会使用到索引，’aaa%’会使用到索引。

下列情况不会使用索引：

1. 如果查询条件中有or，即使其中条件带索引也不是使用索引,换而言之，就是要求使用的所有字段，都必须建立索引。
2. 对于多列索引，不是使用的第一部分，则不会使用索引
3. Like查询是%开头
4. 如果列类型是字符串，一定要在条件中将数据用引号引起来，否则不会使用所有。（添加时，字符串必须’引号’）
5. 如果mysql估计使用全表扫描要比使用索引快，则不会使用索引。

### explain的用法

**explain可以帮助我们在不真正执行某个sql语句时，去分析sql指令。**



### 如何查看索引使用的情况:

show status like ‘Handler read%’;

大家可以注意:

Handler\_read\_key:这个值越高越好,越高表示使用索引查询到的次数

handler\_read\_rnd\_next:这个值越高,说明查询低效

### 大批量插入数据(MySq管理员)了解

对于MyISAM

alter table table\_name disable keys;

loading data //insert语句;

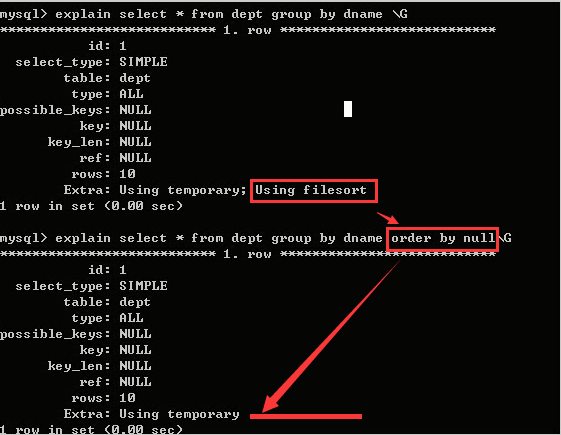
alter table table\_name enable keys;

对于Innodb:

1. 将要导入的数据按照主键排序
2. set unique\_checks=0,关闭唯一性校验。
3. set autocommit=0,关闭自动提交

### 优化 group by语句

1. 默认情况, MySQL对所有的 group by col1,col2进行排序。这与在査询中指定order by col1,col2类似。如果查询中包括group by但用户想要避免排序结果的消耗,则可以使用 order by null禁止排序



1. 有些情况下,可以使用连接来替代子查询。

因为使用join, My SQL不需要在内存中创建临时表。(讲解)

1. 如果想要在含有or的查询语句中利用素引,则or之间的每个条件列都必须用到索引,如果没有索引,则应该考虑增加索引(与环境相关讲解)

Select \* from表名 where条件1=’’ or条件2=’tt’

### MySQL表类型和存储引擎

* 如何选择mysq的存储引擎
* myisam存储:如果表对事务要求不高,同时是以查询和添加为主的,我们考虑使用

myisam存储引擎.此如bbs中的发帖表,回复表

如果你的数据库的存储引擎是 myisam,请一定记住要定时进行碎片整理

举例说明

create table test100(id int unsigned，name varchar(32)engine=myisam

insert into test100 values( 1, ‘aaaaa’);

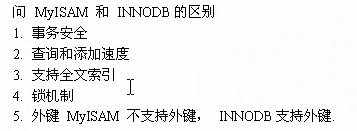
insert into testl00 values(2, ’bbbb’)

insert into test100 values(3, ’ccccc’)

我们应该定义对 myisam进行整理（**清理delete的数据**）

optimize table test100

* INNODE存储:对事务要求高,保存的数据都是重要数据,我们建议使用 INNODB比如订单表,账号表
* memory存储,比如我们数据变化频繁,不需要入库,同时又频繁的查询和修改,我们考虑使用 memory





### 程序备份数据库

1. 手动备份数据库(表的)方法

**cmd控制台**

mysqldump- u root- proot数据库[表名1表名2] > 文件路径

比如:把temp数据库备份到 d:\temp.bak

mysqldump -u root-proot temp >d:\temp.bak

如果你希望备份是,数据库的某几张表

mysqldump -u root -proot temp dept >d:\temp.dept.bak

如何使用备份文件恢复我们的数据

**sql控制台**

source d:\temp.dept.bak

1. 使用定时器来自定完成

把备份数据库的指令,写入到bat文件,然后通过任务管理器去定时调用bat文件

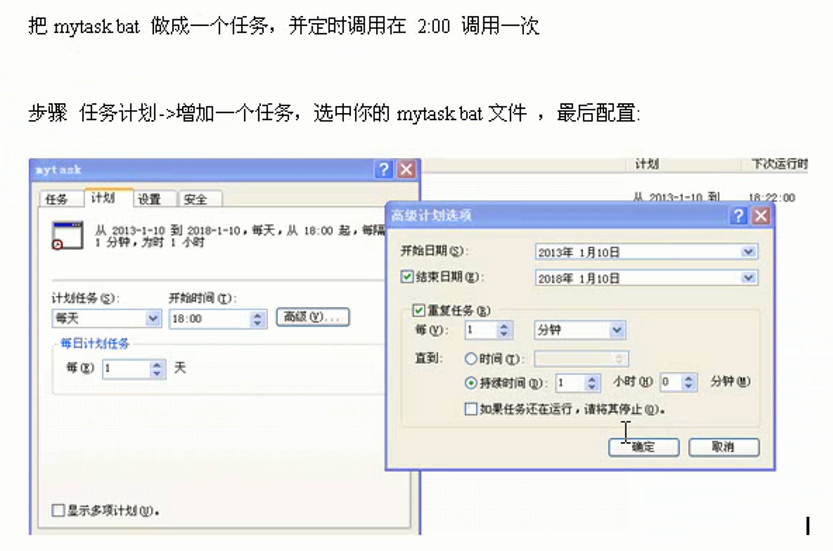
mytask.bat内容是:

C:\myenvmysq15.5.27\bin\mysqldump -u root -proot temp dept > d:\temp.dept.bak

**注意：**如果你的mysqldump. exe文件路径有空格,则一定要使用“”包括

把mytask.bat 做成一个任务，并定时调用在2:00调用一次

步骤：任务计划🡪增加一个任务，选中你的mytask.bat文件，最后配置



## 分表技术（水平分割，垂直分割）

### 水平分割

如果一个表的记录数太多了,比如上千万条,而且需要经常检索,那么我们就有必要化整为零了。如果我拆成100个表,那么每个表只有10万条记录。当然这需要数据在逻辑上可以划分。一个好的划分依据,有利于程序的简单实现,也可以充分利用水平分表的优势。比如系统界面上只提供按月査询的功能,那么把表按月拆分成12个,每个查询只查询一个表就够了。如果非要按照地域来分,即使把表拆的再小,查询还是要联合所有表来查,还不如不拆了。所以一个好的拆分依据是最重要的。UNION

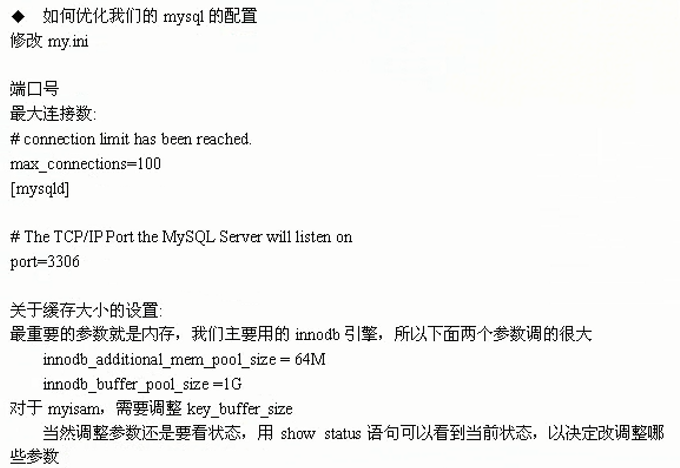
### 垂直分割

把某个表的某些字段,这些字段,在查询时,并不是已经关系,但是数据量很大,我们建议大家可以,把这些字段单独的放到另外一张表,从而提高效率

## 读写【写：update、delete、insert】分离

## 存储过程【模块化编程、执行编译好的sql。 但是移植性不好】

## 对mysql配置优化【my.ini：配置最大并发数，推荐1000左右，调整缓存大小】



## Mysql服务器硬件升级

## 定时清除不需要的数据，定时进行碎片整理。