

## mysql 优化

为查询缓存优化你的查询

查询缓存不开启

```
cursor.execute("SELECT username FROM user WHERE signup_date >= CURDATE());
```

开启查询缓存

```
cursor.execute("SELECT FROM username WHERE signup_date >= %s", [datetime.now()])
```

'%s' % asdfasd

'select \* from user where name = "a";drop database school';

" 'a';drop database school"

sql 注入风险

## EXPLAIN 你的 SELECT 查询

使用 EXPLAIN 关键字可以让你知道MySQL是如何处理你的SQL语句的

## 当只要一行数据时使用 LIMIT 1

判断是否有中国这条记录

```
cursor.execute("SELECT 1 FROM user WHERE country = 'China' LIMIT 1");
```

## 为搜索字段建索引

某个字段你总要学会经常用来做搜索，请为其建立索引

where c =b and a

```
create index ix_tablename_colname on table(col1, col2);
```

## 在Join表的时候使用相同类型的字段，并将其索引

```
cursor.execute("SELECT company_name FROM users  
LEFT JOIN companies ON (users.state = companies.state)  
WHERE users.id = %s", [user_id]);
```

两个 state 字段应该是被建过索引的，而且应该是相当的类型，相同的字符集。

## 避免 SELECT \*

```
SELECT * FROM user WHERE user_id = 1
```

```
SELECT username , dFROM user WHERE user_id = 1
```

## 为每张表设置一个ID

## 尽可能的使用 NOT NULL

## 固定长度的表会更快

例如，表中没有如下类型的字段： VARCHAR，TEXT，BLOB

## 垂直分割

“垂直分割”是一种把数据库中的表按列变成几张表的方法，这样可以降低表的复杂度和字段的数目，从而达到优化的目的。

另外，你需要注意的是，这些被分出去的字段所形成的表，你不会经常性地Join他们，不然的话，这样的性能会比不分割时还要差，而且，会是极数级的下降。

## 拆分大的 DELETE 或 INSERT 语句

如果你需要在一个在线的网站上去执行一个大的 DELETE 或 INSERT 查询，你需要非常小心，要避免你的操作让你的整个网站停止相应。因为这两个操作是会锁表的，表一锁住了，别的操作都进不来了。

```
while (1){  
    每次只做1000条  
    cursor.execute("DELETE FROM logs WHERE log_date <= '2009-11-01' LIMIT 1000");  
    if(mysql_affected_rows() == 0){  
        没有了，退出！  
        break;  
    }  
    每次都要休息一会儿  
    time.sleep(50000);  
}
```

## 选择正确的存储引擎

MySQL 中有两个存储引擎 MyISAM 和 InnoDB

MyISAM 适合于一些需要大量查询的应用，但其对于有大量写操作并不是很好

nnoDB支持“行锁”，于是在写操作比较多的时候，会更优秀。并且，他还支持更多的高级应用，比如：事务。

查看存储引擎类型

```
show create table lesson;
```