哈希算法上：如何防止数据库中的用户信息被脱库?

如何存储用户密码这么重要的数据呢？仅仅用MD5加密一下存储就够了吗?

从实际开发的角度来看，我们该如何用哈希算法解决问题呢?

什么是哈希算法?

哈希算法又叫hash算法或散列算法;

定义和原理：将任意长度的二进制值串映射为固定长度的二进制值串，这个映射的规则就是哈希算法，而通过原始数据映射之后得到的二进制值串就是哈希值。

优秀的哈希算法需要满足几点要求:

1. 从哈希值不能反向推导出原始数据，也称单向哈希算法;
2. 对输入数据非常敏感，哪怕原始数据只修改一个Bit，最后得到的哈希值也大不相同;
3. 散列冲突的概率要很小，对于不同的原始数据哈希值相同的概率非常小;
4. 哈希算法执行的效率要尽量高效，针对较长的文本，也能快速的计算出哈希值。

无论哈希的文本有多长，多短，通过MD5哈希之后，得到的哈希值长度都是相同的。

两个非常相似的文本，“我今天讲哈希算法！”和“我今天讲哈希算法”。这两个文本只有一个感叹号的区别。如果用 MD5 哈希算法分别计算它们的哈希值，你会发现，尽管只有一字之差，得到的哈希值也是完全不同的。

```
MD5("我今天讲哈希算法！") = 425f0d5a917188d2c3c3dc85b5e4f2cb
MD5("我今天讲哈希算法") = a1fb91ac128e6aa37fe42c663971ac3d
```

哈希算法的应用非常多，**常见的有七个：安全加密、唯一标识、数据校验、散列函数、负载均衡、数据分片、分布式存储。**

应用一：安全加密

最常用的加密的哈希算法是：MD5和SHA，以及DES和AES

用于加密的哈希算法来说，有两点格外重要，第一点是很难根据哈希值反向推导出原始数据，第二点是散列冲突的概率要很小。

加密的目的是防止原始数据泄露，所以很难通过哈希值反向推导原始数据，这是一个最基本的要求。 不管什么哈希算法我们都只能尽量减少碰撞冲突的概率，理论上是没有办法可以做到完全不冲突的。

为什么哈希算法无法做到零冲突呢？ 哈希算法产生的哈希值长度是固定且有限的，比如前面的MD5的例子，哈希值固定是128位二进制，能表示的数据是有限的，最多能表示2^128个数据。哈希值越长的哈希算法，散列冲突的概率越低。

应用二：唯一标识

图库中搜索图片，不能单纯的用图片的元信息来对比。可以给每一个图片取一个唯一的标识，或者说信息摘要，通过哈希算法得到一个哈希字符串，用他来作为唯一标识，从而来判定图片是否存在于图库中。

应用三：数据校验

电驴这种BT下载软件，BT下载原理是基于P2P协议的，从多个机器中并行下载一个2GB的电影，这个电影文件可能会被分割成很多文件块，比如可以分成100块，等所有文件块都下载完成后，在组装成一完整的电影。

网络传输不安全，下载文件块可能被宿主机恶意修改过，又或者下载过程中出现错误，所以下载文件块可能不是完整的。 所以如何校验文件块的安全，正确，完整呢?

通过哈希算法，对100个文件块分别取哈希值，并且保存在种子文件中，当文件下载完成之后，我们可以通过相同的哈希算法，对下载好的文件块注意求哈希值，然后跟种子文件中保存的哈希值对比，如果不同说明这个文件块不完整或者被篡改了，需要重新从其他宿主机下载这个文件块。

应用四：散列函数

散列函数是设计一个散列表的关键。他直接决定了散列冲突的概率和散列表的性能。相对哈希算法的其他应用，散列函数对于散列算法冲突的要求要低很多。即便出现个别散列冲突，只要不是过于严重，我们都可以通过开放寻址法或者链表法解决。

解答开篇

通过哈希算法，对用户密码加密之后在存储，最好选择相对安全的算法，比如SHA等。但仅仅这样就安全无事了吗?

字典攻击，用户信息脱库，黑客虽然拿到的是加密之后的密文，但可以通过猜的方式来破解密码，这是因为，有些用户的密码太简单，比如很多人习惯用00000,123456等简单组合来做密码，很容易被猜中。

针对字典攻击，可以引入一个盐salt，跟用户的密码组合在一起，增加密码的复杂度。我们拿组合之后的字符串做哈希算法加密。

课后思考

现在，区块链是一个很火的领域，它被很多人神秘化，不过其底层的实现原理并不复杂。其中，哈希算法就是它的一个非常重要的理论基础。你能讲一讲区块链使用的是哪种哈希算法吗？是为了解决什么问题而使用的呢?

区块链是一块块区块组成的，每个区块分为两部分：区块头和区块体； 区块头保存着自己区块体和上一个区块头的哈希值。 因为这种链式关系和哈希的唯一性，只要有一个区块被修改过，后面所有区块保存的哈希值就不对了。 区块链使用的是SHA256哈希算法，计算哈希值非常耗时，如果篡改一个区块，就必须重新计算该区块后面所有的区块的哈希值，短时间内几乎不可能做到。