# 第二阶段设计文档

这一阶段主要任务是实现两个命令：

1. [encrypt](C:/Users/door/AppData/Local/youdao/DictBeta/Application/7.0.0.2017/resultui/dict/../dict/result.html?keyword=encrypt) table（表名） [property](C:/Users/door/AppData/Local/youdao/DictBeta/Application/7.0.0.2017/resultui/dict/../dict/result.html?keyword=property)（属性名） --------加密，实质是设置敏感数据
2. decrypt table（表名） [property](C:/Users/door/AppData/Local/youdao/DictBeta/Application/7.0.0.2017/resultui/dict/../dict/result.html?keyword=property)（属性名） -------解密

## [encrypt](C:/Users/door/AppData/Local/youdao/DictBeta/Application/7.0.0.2017/resultui/dict/../dict/result.html?keyword=encrypt) table [property](C:/Users/door/AppData/Local/youdao/DictBeta/Application/7.0.0.2017/resultui/dict/../dict/result.html?keyword=property)

加密，实际是设置敏感数据——该命令对整个属性（列）进行加密，并且自此后一直机密，除非利用命令（2）对其进行解密。

注：这里只对加密命令作具体解析，解密命令decrypt table [property](C:/Users/door/AppData/Local/youdao/DictBeta/Application/7.0.0.2017/resultui/dict/../dict/result.html?keyword=property)其实就是他的逆反。

## 实现流程：

1. 确保表和列的存在

表：select \* from sys.tables这个命令会把所有的表都列出来（本来是想把我所用的数据库的表列出来就行）——先凑合着用。

列：select name from syscolumns where id = object\_id('graduation\_project.dbo.user\_tb')——查询所有的列（属性），graduation\_project——数据库，dbo——数据库所有者，user\_tb——表名。

感受：个人认为一定会有更好的方法实现这个功能，对于目前方法我们先使用着，以后可以作优化。

1. 取得加密的密钥和向量

密钥和向量都是长度为8的字符串

1. 取得对应表和对应列的数据进行加密

三步：取（SELECT）---->加密（SELECT）---->更新（UPDATE）

1. 记录加密信息（message\_tb）

感受：步骤还是较多的，但是很容易理解

## 关于加密解密的几点思考

1. message\_tb记录加密信息，那么这个表的重要性不言而喻，但是目前来说并没有对这个表作安全性操作，例如：加密、加权限等。。
2. 关于多线程访问数据库的问题，它们是共享一个真实用户，我可以认为是sql语句是顺序执行（这个应该怎么说呢，好像即使是多个真实用户也是顺序的），读者可以忽略我前面的表达，总的来说，我考虑就是数据错误的问题，大家可以想一下如果加密解密同时进行，应该就会造成数据错误。
3. 为了避免数据错误，采取加锁和事务的方法。那么每个事务都是隔离开来的，这里加密和解密分别是两个事务，我们以后都采取这种实现方式。

## 命令处理的线程实现模式

简单点说，就是一个命令处理一个线程，这样就实现了命令同时处理的功能了，这可以加快程序的速度。（这个好理解，不做解析了）