在接触了比特币和区块链后，我一直有一个想法，就是把所有比特币的区块链数据放入到关系数据库（比如SQL Server）中，然后当成一个数据仓库，做做比特币交易数据的各种分析。想法已经很久了，但是一直没有实施。最近正好有点时间，于是写了一个比特币区块链的导出导入程序。

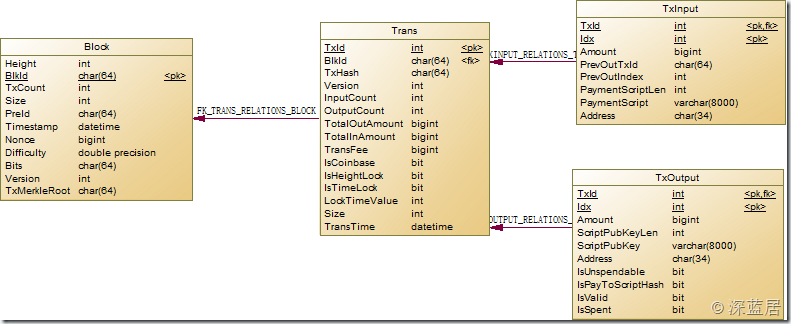
之前我的一篇博客：[在区块链上表白——使用C#将一句话放入比特币的区块链上](http://www.cnblogs.com/studyzy/p/write-string-to-blockchain.html)  介绍了怎么发起一笔比特币的交易，今天我们仍然是使用C#+NBitcoin，读取比特币钱包Bitcoin Core下载到本地的全量区块链数据，并将这些数据写入数据库。如果有和我一样想法的朋友，可以参考下面是我的操作过程：

## **一、准备**

我们要解析的是存储在本地硬盘上的Bitcoin Core钱包的全量比特币数据，那么首先就是要下载并安装好Bitcoin Core，下载地址：[https://bitcoin.org/en/download](https://bitcoin.org/en/download" \o "https://bitcoin.org/en/download) 然后就等着这个软件同步区块链数据吧。目前比特币的区块链数据大概130G，所以可能需要好几天，甚至一个星期才能将所有区块链数据同步到本地。当然如果你很早就安装了这个软件，那么就太好了，毕竟要等好几天甚至一个星期，真的很痛苦。

## **二、建立比特币区块链数据模型**

要进行区块链数据的分析，那么必须得对区块链的数据模型了解才行。我大概研究了一下，可以总结出4个实体：区块、交易、输入、输出。而其中的关系是，一个区块对应多个交易，一个交易对应多个输入和多个输出。除了Coinbase的输入外，一笔输入对应另一笔交易中的输出。于是我们可以得出这样的数据模型：

[](http://images2015.cnblogs.com/blog/2625/201706/2625-20170618234528400-802220421.png)

需要特别说明几点的是：

1.TxId是自增的int，我没有用TxHash做Transaction的PK，那是因为**TxHash根本就不唯一**啊！有好几个不同区块里面的第一笔交易，也就是Coinbase交易是相同的。这其实应该是异常数据，因为相同的TxHash将导致只能花费一次，所以这个矿工杯具了。

2.对于一笔Coinbase 的Transaction，其输入的PreOutTxId是0000000000000000000000000000000000000000000000000000000000000000,而其PreOutIndex是-1，这是一条不存在的TxOutput，所以我并没有建立TXInput和TxOutput的外键关联。

3.对于Block，PreId就是上一个Block的ID，而创世区块的PreId是0000000000000000000000000000000000000000000000000000000000000000，也是一个不存在的BlockId，所以我没有建立Block的自引用外键。

4.有很多字段其实并不是区块链数据结构中的，这些字段是我添加为了接下来方便分析用的。在导入的时候并没有值，需要经过一定的SQL运算才能得到。比如Trans里面的TotalInAmount,TransFee等。

我用的是PowerDesigner，建模完成后，生成SQL语句，即可。这是我的建表SQL：

IMG_257 View Code

## **三、导出区块链数据为CSV**

数据模型有了，接下来我们就是建立对应的表，然后写程序将比特币的Block写入到数据库中。我本来用的是EntityFramework来实现插入数据库的操作。但是后来发现实在太慢，插入一个Block甚至要等10多20秒，这要等到何年何月才能插入完啊！我试了各种方案，比如写原生的SQL，用事务，用LINQToSQL等，性能都很不理想。最后终于找到了一个好办法，那就是直接导出为文本文件（比如CSV格式），然后用SQL Server的Bulk Insert命令来实现批量导入，这是我已知的最快的写入数据库的方法。

解析Bitcoin Core下载下来的所有比特币区块链数据用的还是NBitcoin这个开源库。只需要用到其中的BlockStore 类，即可轻松实现区块链数据的解析。

以下是我将区块链数据解析为我们的Block对象的代码：

IMG_258 View Code

至于WriteBitcoin2Csv方法，就是以一定的格式，把Block、Trans、TxInput、TxOutput这4个对象分别写入4个文本文件中即可。

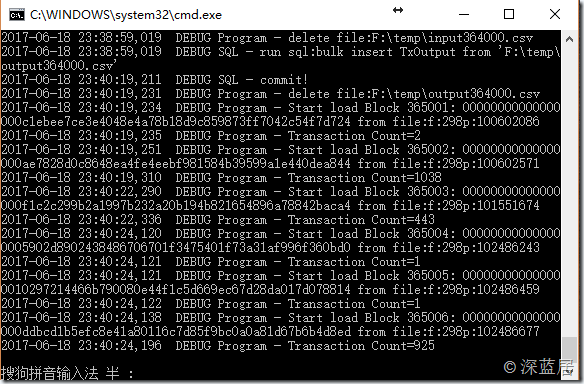
## **四、将CSV导入SQL Server**

在完成了CSV文件的导出后，接下来就是怎么将CSV文件导入到SQL Server中。这个很简单，只需要执行BULK INSERT命令。比如这是我在测试的时候用到的SQL语句：

bulk insert [Block] from 'F:\temp\blk205867.csv'; bulk insert Trans from 'F:\temp\trans205867.csv'; bulk insert TxInput from 'F:\temp\input205867.csv'; bulk insert TxOutput from 'F:\temp\output205867.csv';

当然在实际的情况中，我并不是这么做的。我是每1000个Block就生成4个csv文件，然后使用C#连接到数据库，执行bulk insert命令。执行完成后再把这生成的4个csv文件删除，然后再循环继续导出下一批1000个Block。因为比特币的区块链数据实在太大了，如果我不分批，那么我的PC机硬盘就不够用了，而且在导入SQL Server的时候我也怀疑能不能导入那么大批量的数据。

最后，附上一张我正在导入中的进程图，已经导了一天了，还没有完成，估计还得再花一、两天时间吧。

[](http://images2015.cnblogs.com/blog/2625/201706/2625-20170618234530196-1956152046.png)

所有区块链数据都进入数据库以后，就要发挥一下我的想象力，看能够分析出什么有意思的结果了。