Winston Shine

ILC Database Theory/Security

Week 7 Progress Report

2/28/2023

My goal is to study Database Theories and Security through the book Translucent Databases. Some additional smaller goals I am trying to accomplish are doing my work this quarter with docker containers and If time allows study a bit about document-oriented databases.

This week I was continuing to study Quantization. I did some more reading trying to understand adaptive quantization which I talked about in my week 6 progress report. Adaptive quantization is a scheme where you take a set of data and separate it into groups, then one value is designated to replace/represent any data that is a part of that group. There are many ways to determine the groups like splitting the groups into equal sizes, or attempting to group similar data. The example given where adaptive quantization is heavily used were in compression algorithms.

Projection is another method of quantization I read about that tries to take a dataset of a higher dimension and represent it in a meaningful way in a lower dimension. One method used to accomplish this is singular value decomposition, the formula for which is M = UDV^t. M is the original data set and D is a diagonal matrix of singular values. The number of values along the diagonal matrix D corresponds to the number of dimensions of the original data set. To effectively reduce the number of dimensions, you would replace some of the values in D with 0s, the number of values not zero would equate to the number of dimensions in the result. There are still pieces I don’t understand but I started to get a better grip on the math thanks to the explanation of singular values on <https://www.omnicalculator.com/math/singular-values> .

The simplest way to use quantization in a database is to discard the left over or unneeded data, but often times you will want to keep that data but only make it available to specific users. Two methods of this are to encrypt pieces data, or add random noise. Adding random noise would be changing a value or adding extra parts to it in order to obscure the original value. The example given was working with values of time where the minutes and seconds would be changed systemically using modular arithmetic so the end values would still look like minutes and seconds, but would not actually reflect the original values. I believe for this method to be useful the method used to add noise would need to be reversible but that is one thing I did not quite understand in this section. Or maybe whether it is reversible or not is just a design decision.

Overall quantization in databases is about either removing or obscuring some details while leaving others how they are. This adds security by keeping important information hidden while allowing the rest of the data to be used. A good example the book Translucent Databases provided was if you were supplying researchers with data to analyze they might only need some of the information, and if the data was about people you would not want to give the researchers their names or location they live so those could be obscured or removed from what is provided to the researchers while leaving information that was relevant. (The example given was data on the smoking habits of people in correlation to the amount of radon in their house and location).

Next week I plan to do some practice writing queries that can enter encrypted or obscured information into a database. For the reading next week I plan to read the next two chapters titled ‘Coordinating Users’ and ‘Synchronization’.