Sean Connor Discussion Prompt 10 7 August 2018

1. What are some of the largest employers with which you are familiar? What do you imagine are some of the greatest challenges facing them in terms of information management?

I currently work for Johns Hopkins School of Public Health, which falls under the larger Johns Hopkins system. JHU is a massive employer in the Baltimore-Washington area. Managing a large amount of employee records, patient records, student records, inventories, etc. requires an efficient and functioning system. The greatest challenges I feel are likely to be the sheer volume, and keeping all of the different records organized for quick and efficient retrieval of data. Health data security is also a big issue.

2. What are some of the many ways that company may need to access data?

A company may need to access data through a software program, through a web portal, through a database, through traditional paper filing. In my small research lab, we primarily use Excel has a platform for managing our data, to include everything from results of experiments to freezer inventories. However, this definitely has drawbacks and we have been looking into implementing a different system.

3. What are the pros and cons of using a social security number as a primary key? What are some good alternatives?

Pros: They are unique to each person.
Cons: The first several digits are dictated by location of birth, and so the first several digits are identical for many thousands of records. This can make using social security numbers as a key not practical in many cases. Alternatives could be name (i.e. lastfirstmiddle), date of birth (i.e. 19690720), or some combination of the above (i.e. doejohnsmith19690720).

4. How big an impact, on your design considerations in general, is it to need a sorted file only periodically (for a directory, for example)?

I would say not a huge impact. In general I would try to optimize all applications, unless the time constraints/difficulty is too high. As such, I would seek to implement the most efficient sort algorithms from the start. Frequent sorting or infrequent sorting – the algorithm I choose would likely be the same.

5. How big an impact on, your design considerations in general, is it to sort by different fields of the data periodically (a directory is likely ordered by last name, first name, for example)?

See response to question 4 above. I’m not sure I see how this is different. In either case, one is sorting by some field (key, last name, etc).

6. How would the nature of your data set affect your choice of collision handling scheme?

Generally, the schemes are either chaining or open addressing. Chaining is able to handle load factors greater than 1, but can grow out of of size which could limit its possible applications. Open addressing does not have this problem, but also cannot handle load factors greater than 1. So I suppose that the nature of the data set would probably not affect my choice of collision handling scheme, but the intended application could.