Elliot Smith
Research 1 - RDBMS

1. At the time of this paper's origination, applications that interface with databases were required to understand the precise structure and organization of the underlying data; this resulted in tight coupling between the database and the applications accessing them. As a result, as data stores needed to evolve as the needs of businesses changed, all of the applications that accessed the data would need to be updated as well to work with the new data structure. The need for the relational database approach was so that applications that access the data store would not have to understand "the internal representation" of the data to perform queries and database interactions.

2. The advantage that the relational representation had over the day's solution was its ability to completely decouple the data store and the applications that needed to access them. Prior to the relational model, an application needed to have a deep understanding of the underlying data model, and as such, would be required to update the mechanics of the application if the database went through structural changes. With the relational model, however, there were no absolute paths, meaning that if the structure of the data changed it would have no impact on existing application workflows.

3. The key object here is data storage that facilitates independence between the database itself and any interacting applications. If an application is built to access a data store, the application should not need to make any programmatic changes to its logic if that data store makes any material, structural changes. The author intends to show that his method of data storage is superior than the current method as it deals with this intrinsic failing that existed in 1970.

4. A lot of the implementation details were left out by the author and as such the paper was mostly theoretical in nature, though it was supported by conventional algebra that made the information easier to synthesize. One of the key challenges in my mind (and from previous work experience) is dealing with database insertion or update issues. For example, one of the key tenets of the authors design is that all rows are distinct values. How do we handle an event where a non-distinct tuple is being added to a relation? The trade-off of checking for issues at insertion time would better preserve the data integrity while slowing down the proficiency of those tasks. While on the contrary, a nightly batch job, as a counterexample, would handle these issues during off-hours but the real-time data integrity may be suspect as a result.

5. The key insight is the need for independence between a data store and any applications that interact with it for both improving existing technology and discovering new approaches to solve complex problems. As a result of this principle of data independence inherent in the relational model, we have seen the rise of third party

applications that have the ability to interface with a wide variety of different data stores; a manifestation that would have been impossible without the independence between database and application. This fact has led to some very fascinating discoveries in the world of computer science. Previously, application developers would need to have a very strong understanding of the data store on which they were developing; however, now that a wider variety of users can build applications on top of databases, we are seeing an exponential advancement in many areas. In addition, the relational model makes it easier for non-associated developers (individuals who are not employees of an organization) to develop on existing data stores and innovate.

6. The idea that certain computer science methodologies can have foundation in theoretical mathematics is very novel to me and I can certainly see the appeal. As a Statistics Masters student, so much of my work is predicated on explaining my assumptions and proving the my approach has sound logic in the face of potential criticism. The idea that the author can have a mathematical basis for his theory allows him to both show that his concept are theoretically sound and have “conceptual parity” that will allow a technical lay person to understand the theory without knowing what the practical application is. This enhances his position, and undoubtedly, the field itself, as future computer scientists will use this example as a template for how to prove their points both practically and theoretically.

7. Atzeni P, Jensen C, Orsi G, Ram S, Tanca L, Torlone R. The relational model is dead, SQL is dead, and I don’t feel so good myself. SIGMOD Record, June 2013 (Vol 42, No 2).

   This paper acts as a contemporary pseudo-rebuke of the relational model in today’s current data setting. While the original paper by EF Codd talked about the advantages of the relational model in a setting where data independence was a pressing issue that needed a resolution, this subsequent paper addresses the fact, that in many disciplines, the needs of the data-consuming world have changed in such a way that the relational model is no longer the default choice. The authors discuss the advantages of unstructured data storage that is more advantageous in today’s environment where the types of data are no longer the prototypical administrative applications with well-structured data. They further discuss the need to store data that is unstructured, continuous, sensor data, streaming data and the like that a relational database is not equipped to handle in an efficient manner.