Name : Zhenyu Fan

Class : CSCI 4707

Instructor : Shana Walter

Date : 19/11/2018

# Review of Codd Paper

This paper of relational database is ground breaking in that period of time and had transformed the way people store and use data ever since. Before the idea of relational database coming up, most of the database systems are arranging the data in an order way that could not only raise numerous error when used but also inconvenient for both the users and platforms.

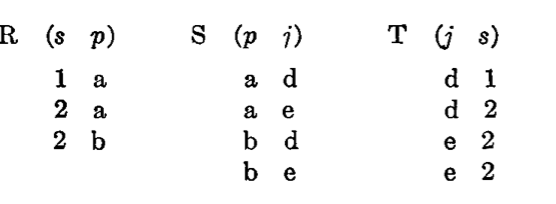
During the old times, the database are mostly order in same way because it is more straightforward at the time. This straightforward implementation of data storage, thought intuitive, could mess up many programs. For example, this model stores the data row by row, column by column in a fixed order. Then, whenever a user or a program wants to use the data in the database, they have to make a request in the same order as the way it is stored in the database. Also, this model is index dependent and address dependent, and therefore would both make the program inefficient because of storing additional index information and erroneous if the internal structure of data is changed.

E. F. Codd, a researcher of IBM, saw this potential risk that exist in that old-time database system and therefore proposed a ground-breaking new model of data storage, the relational data model. One of the biggest innovation of this model is that it eliminates many dependancies from the old model. In relational database model, each row will represent a distinct piece of information. While each row of this relational model is not arranged in any order, each column of this model is labeled and represent the nth element of each piece of information. In this way, we can retrieve the data that we desire simply by requesting the data of the specific label.

Another assumption E. F. Codd made to solve the weakness of the old-time database system is to introduce the idea of normalizing forms. In the old times, many database has multi-values attributes. For example, if the names of my ex-girlfriend is stored in the old time database, it would be likely that my information is an entity and all my ex-girlfriend’s name are crammed into one slot. Though this is easy to implement for data recorders, It would be extremely hard for us to extract a specific ex-girlfriend and possibly use this ex-girlfriend’s information to dig up more information. It would be just so messy and inconvenient for us to utilize a piece of information that is stored in such multi-values attributes.

Therefore, E. F. Codd proposed the idea of normalization. Instead of cramming multiple values together to form a non-atomic “big value”, he introduced the new way of creating a new relation with each element of big value as an atomic value of the new relation and with the identifying key, primary key, of the parent relation as a foreign key to connect the relations. In my upper example, it would look like my\_ex-girlfriend relation have all the girl’s name as attributes and my SSN so that we could link these two relations together and use the data in a clearer and more efficient way.

Overall, E. F. Codd presents a very good model of data science, however, there are a little confusion made when he is talking about primary key and joining. Theoretical, every row of the data should be unique at least in its relation. However, the author gave us an example like this.



with R, S, T as a table that is derived from the original table. But I noticed that there are two rows in T with the same value of (e,2). So I wondered is this an overlooked mistake or there is something that the author missed since the relational database model is a brand new thing at the time.

If I was a cutting-edge data scientist in that times and have never seen any other database system like relational data model, I would be so impressed by this model. Because even if I am a supporter for the traditional way of storing data, I should have noticed drawbacks of old model. I would never be able to forget about the orderings of the data, the address of the data structure and etc and therefore would never be able to process the data that is present in my database so freely. It would be difficult for me to accept such an unprecedented model but I can still see the potential that lies within this theory and I will definitely accept and dedicate more time into developing this model .

No matter what kind of database I am used to, I would be so impressed about the domain-unordered counterparts that used in the relationships. Users can be freed from burdens with remembering the domain ordering of any relation. To replace the orderings and make this model work, we simply need to assign domain names to the specific domain and make sure that every rows of the data that we put into our database are distinct to each other. Once we have the relational model set up like this, we can freely access the specific domain of the data even out ever remembering what index they are. However, we also need to ensure the uniqueness of the rows and the coordination between relations. This is when the primary key and foreign keys comes in. Primary key is the identifying element that exist in each row and every single row has their own distinct primary key and thus the uniqueness of the rows in ensured. Then, we will need foreign keys. Foreign keys are the elements in a relation that points to the primary keys of other relations and thus we can link 2 relations together and use all the relation tables together.

This particle is excellent except that there is one thing that I cannot understand. When the author is talking about the strong redundancy, and mention a formula

I am so lost because I have no idea where this formula came from and what it represent. Therefore, I learned nothing from this redundancy section. I guess it is just too hard for me without enough math and database background knowledges.

This article is the beginning of the relational database system that is massive used in our daily lives. Even if there are some confusions and probably error in this article, it definitely push the relational database system into the stage of computer science. This article is the reason why the later generations can construct a more solid system on the relational database model and I definitely benefit a lot from it.