

Codd splits the paper into two sections: the first section explaining the inadequacies of the current data systems and the second section discusses how relational operations are applied to the problems of redundancy and consistency in the user's model in order to maintain a consistent data state. The main issue is to be able to alter the data representation without impairing the application programs that were currently running.

Codd begins by listing the current problems with the data dependencies in present systems. He states that the three principal kind of data dependencies that need to be removed are ordering dependence, indexing dependence, and access path dependence. Each of these dependencies have their own purposes, but they all have a downside. For instance, the access path dependence provides tree structured files or network models of data which can impair application programs that work with these. The solution to avoid this is that once a user access path is defined, it shouldn't be made obsolete until all applications using the path are also obsolete. However, it's impractical as the size of the data bank will eventually become massive. Codd also explains that there is an issue with dealing with data columns and domains that have identical names as well as users dealing with relationships rather than remembering the domain ordering of any domain.

Codd then explains several relational operations, that are notably used for their key roles in deriving relations from other relations, as well as two types of redundancy and how they are applied in order to maintain a consistent data state. The relational operations that are applied include permutation, projection, join, composition, and rejection. Permutation is used on binary relations where their columns are interchangeable. Projections are an array of select relation columns which have no similar rows. Join includes two binary relations which both have some common domain. Two relations are composable if and only if they are joinable. Lastly, when a subset of a relation is a relation, this is known as restriction. These relational operations are used since relations are essentially sets, but the result of these operations are not usually relations.

Codd also explains the two different types of redundancies: strong redundancy and weak redundancy. Strong redundancy is a set of relations that contains at least one relation which has a projection that is derivable from other relation members of the set. Weak redundancy, on the other hand, is nearly the opposite as it contains a relation that has a projection which cannot be derived from other members.