

A database that most of us interact with on a daily basis is Marist's CWID system. The system stores raw data (Alliyah, Faith, 200-YY-YYY, Taylor, 200-XX-XXX, Matthew), which alone, is meaningless. Matthew could be a first name, a last name, or a random string of letters that resembles both. The database identifies each piece of data: The CWID, the First Name, and the Last Name, and matches them with the appropriate student's information. In this database, the CWID is the unique entity with first and last names as its characteristics. This way, even students who share the exact same first and last name are treated as individual students by the system. When given context, we know not only that Matthew is a last name, but also that it is the last name of the student Faith Matthew, whose CWID is 200-XX-XXX.

The hierarchical database model organizes data into a “family tree” like structure with “children” and “parents”. Each “child” has one “parent”, but a “parent” may have multiple “children”. In this model, you must go through the whole tree from the top to retrieve any data. In this model, parents cannot share children. The network model improves upon this, allowing the “children” in the tree to have multiple “parents”. This helps to reduce redundancies within the model. The relational database takes this even further. By representing the data as tables and defining relationships between these tables, there is no need for redundant entries in the database. By going back from a tabled structure to a tree structure, XML feels more clunky than a relational database. It’s readability makes it attractive, but only when the people who need to access the database don’t have the understanding they need of a relational database.

