# A COMPARISON OF THE RELATIONAL DATABASE MODEL AND THE ASSOCIATIVE DATABASE MODEL

Joseph V. Homan, Robert Morris University, jvhst4@mail.rmu.edu
Paul J. Kovacs, Robert Morris University, kovacs@rmu.edu

================================================================================

## ABSTRACT

*This paper compares the relational database model with the associative database model. This paper briefly summarizes the relational and other familiar data models. The remainder of this paper will introduce and describe the associative database model. The associative model is less known because it is relatively new and does not have a large software supplier base. While it seems to offer a number of benefits and advantages over other database structures, it has yet to become a commercial success in the mainstream database market.*

**Keywords:** associative databases, the associative database model, database models, database modeling, database relationships.

## INTRODUCTION

Information technology professionals engaged in database design or architecture are probably familiar with the relational database model. This paper will briefly summarize the relational and other familiar data models. The remainder of this paper will introduce and describe the associative database model. The associative model is less known because it is relatively new and does not have a large software supplier base. While it seems to offer a number of benefits and advantages over other database structures, it has yet to become a commercial success in the mainstream database market. However, given the growth of larger and larger databases and data warehouses – and the speed and efficiencies promised by the associative data model – this data model is likely to be a major new addition to the database market. This paper will primarily focus on an introduction of the associative data model and comparison to a relational data model – with limited examples of benchmark data or real-life applications. However, a series of subsequent papers are planned to describe the merits of this new data model and actual applications of the technology. Note: for the purposes of this paper, database management system (DBMS), data model, and data architecture – although not synonymous – are used somewhat interchangeably.

## DATABASE HISTORY

McLeod and Schell [13] note that "database management systems have a relatively short history" with the first DBMS being "developed by GE in 1964." As described by Simon [16], mainframes were the source of early data and data processing applications. As the mainframes began to be replaced with minicomputers, people started to look for ways to gain access to data more directly rather than making "requests to the data-processing department every time data from an application's files or databases" were needed. To get more direct access, standard data resulting from a query would be copied to a tape for transfer to and use on a minicomputer. This approach worked in some cases, in other cases it was problematic due to the time lag and need to re-query the mainframe if any different data were needed.

In the 1980s, microcomputers and computer applications made distributed computing widespread. Although efficient distributed database management systems were envisioned, "islands of data" was the problem that these PC-based applications created [16]. From 1990 to the present, the advancement in more powerful hardware, software, and network technology – including the advent of the Internet and related global telecommunications capability – resulted in powerful database software, huge data repositories and warehouses, and a variety of models and structures for storing data. The underlying goal in all of this growth and advancement of technology was not just for the sake of technology. The goal was to make data available to people – in the right form, in a timely manner, and the right context – to allow them to make appropriate decisions about the data related to their business or function. To do that, the data must have meaning, be converted to information and enhance or create knowledge for the user.

## MOVING FROM DATA TO INFORMATION TO INTELLIGENCE AND KNOWLEDGE

Data are separate bits of information with no related context to give them meaning. For example, this string of data elements – #PM764, 60, SN 1239876a, 814-555-1212 – each represent something, but are meaningless with no context or connection to other data. Data are meaningless without a relationship (e.g., 60). If you take any of these data elements and combine it with other information or show a relationship, you get information, i.e., pieces of data with relationships. For example, if you indicate 60 employees, Employee ID Card #PM764, and phone number 814-555-1212, then each piece of data becomes more meaningful. Any piece of information is relatively meaningless without context. For example, if you say Pat Smith is 60 – is 60 an age, a weight, or an IQ? Context gives information relevance. For each context in which information is represented, it may have a unique set of relationships to other pieces of information. There are as many types of relationships as there are ways of thinking about things. Information, in any context, can have an infinite number of relationships to other information elements or sets. Information can be gathered in an infinite number of sets and information elements can exist simultaneously in any number of different information sets. Information grows by adding perspective and intent.

To take information to the level of intelligence, it must be further contextualized**.** As indicated previously, data refers to characters, numbers or images without context and by itself has no meaning. Data becomes information when the data are processed within a context**.** Intelligence is purposed information. Using the data and information for the previous example, one can contextualize it as follows: EID Card #PM764 belongs to an employee named Pat Smith who entered Building 2 at 8:17am on 1/2/2009. Taking that concept one step further, you can convert that intelligence to knowledge by adding perspective, history, and broader global context as follows: employee ID cards have embedded codes which indicate the holders' security clearance, can act as tracking devices within the building perimeter, and can prevent the holder from logging into restricted databases above their clearance level. Knowledge is beyond the ability of today's computational capabilities, but can be enabled by making available the appropriate data, information, and contextualized intelligence.

The introduction, history and explanation of moving from data to information to intelligence and knowledge are important to understand the importance of database modeling. So how does one organize, model, and store data to give you information, intelligence, and knowledge? To achieve this goal, a number of database models have been developed. Four that will be described in the following pages include: hierarchical, network, relational, and associative. The majority of this paper will focus on the last two.

## HIERARCHICAL AND NETWORK DATABASE MODELS

Two of the relatively well-known database models are the hierarchical and network models. Hierarchical models involve "a data structure in which the elements of the structure have only one-to-many relationships with one another" [11]. Figure 1 below shows a simple representation of a hierarchical database model.

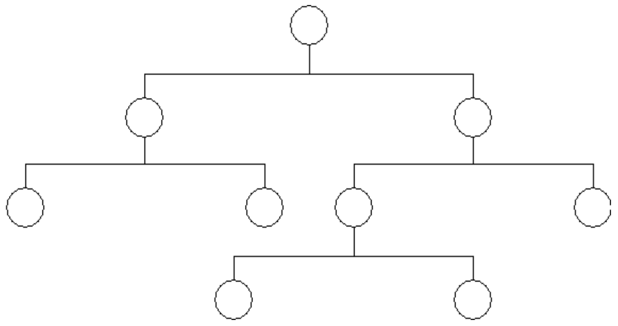

**Figure 1 Simple Hierarchical Database Model**

Similarly, Kroenke [11] defines a network database structure is one "in which at least one of the relationships is many-to-many." Figure 2 depicts a simple network database model.

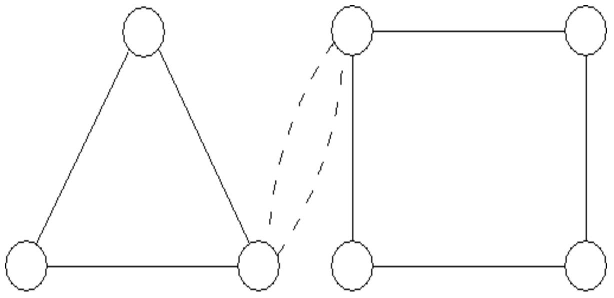

**Figure 2 Simple Network Database Model**

## RELATIONAL DATABASE MODEL

One of the most commonly used database structures is the relational database model. As the name implies the relational database model "has the relation at its heart, but then a whole series of rules governing keys, relationships, joins, functional dependencies, transitive dependencies, multi-valued dependencies, and modification anomalies" [14 The Relational Data Model, para. 1]. Figure 3 below depicts a number of tables showing elements like table names, primary and secondary keys, field names, and relationships.

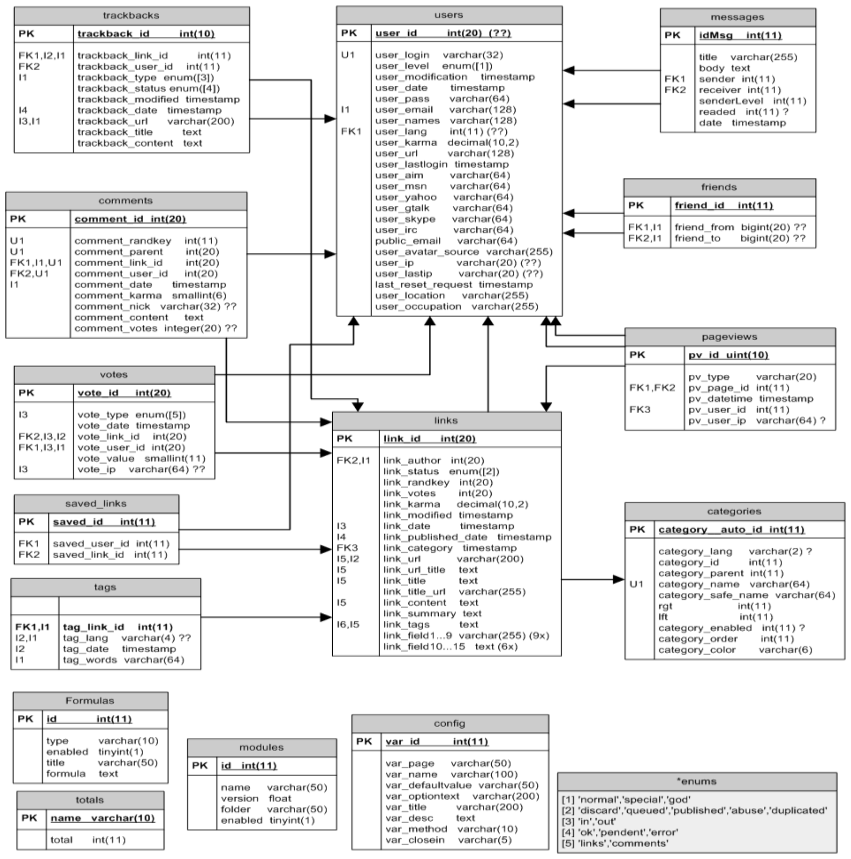

**Figure 3 Relational Database Tables Showing Relationships**

While figure 3 appears to be very complex, a more simple way of looking at the relational database model is show in figure 4. This figure illustrates a simple table view, commonly seen in a spreadsheet application.

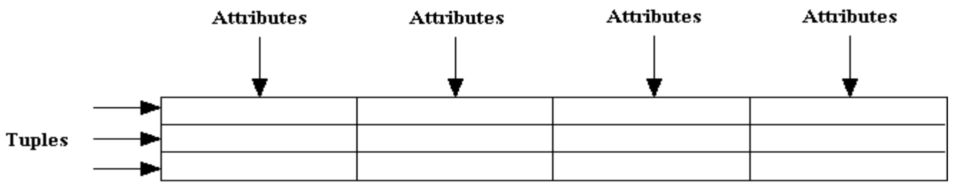

**Figure 4 Simple Relational Database Table**

Figure 5 populates the table structure with data to show both the attributes (header row at the top of the table) and the Tuples, or rows which represent the data in the database.

| STUDENT | COURSE | INSTRUCTOR | HOUR | ROOM | GRADE |
|---|---|---|---|---|---|
| Matt | Math | Steen | 8:00 | 2B | A |
| Mark | History | Scarpino | 8:00 | 4A | B+ |
| Luke | Science | Kim | 8:00 | 3C | C- |
| John | Math | Steen | 11:00 | 2B | A |

**Figure 5 Sample Populated Relational Table**

Relational database management systems (RDBMS) are widely used and have a long history. The recognized 'father' of the relational database, E. F. Codd, wrote an often quoted introduction to an article "[f]uture users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation)" [4]. Textbooks in the early days of database history touted the future of the RDBMS. Kroenke [10] noted that while other early database "models tend to add complexity as they force the user to formalize his view of the data; the relational model tends to simplify." Later, as use of RDBMS became more widespread, the complexity associated with design of a RDBMS was also well documented. Jordan and Machesky [8] identify two key players in the development of a database when they state "the systems analyst must still be aware of general principles of database design in order to help the DBA design a high-quality database." More recent guides for development of databases using specific database packages have similar cautions and directions. Dewson [6] summarizes by saying "[d]esign is not an area that can be skipped or taken lightly" and Rankins, et al. [15] indicate that "[d]esigning for performance requires making trade-offs … to get the best write performance out of a database, you must sacrifice read performance."

Normalization is a major challenge in RDBMS design. Kroenke [10] defines normalization as "a process for converting a relation that has certain problems to two or more relations that do not have these problems." Much is written about tips for normalizing a database and many authors note that an RDBMS is not the best database design for all types of data. Even in more recent writings by the father of the RDBMS, Codd [5] admits that a "relational database is best suited to data with a rather regular or homogeneous structure" and that more research is needed to determine if an RDBMS can sufficiently handle "heterogeneous data" such as "images, text, and miscellaneous facts." Perhaps other database models are more suited to such heterogeneous data.

## ASSOCIATIVE DATABASE MODEL

The associative database model is claimed to offer advantages over RDBMS and other database models. While an RDBMS and other database models are record-based – with data stored in rows and columns in tabular representations shown in figures 4 and 5 – all data in the associative database model is modeled as discrete independent data elements. Relationships between data elements are modeled as associations. Griffiths [7] describes these "two fundamental data structures" as "'Items' and a set of 'Links' that connect them together." Williams [17] introduces the associative database model as having two types of data structures, Items which have "a unique identifier, a name and a type" and Links which have "a unique identifier, together with the unique identifiers of three other things, that represent the source, verb and target of a fact that is recorded about the source in the database." He further notes that "each of the three things identified by the source, verb and target may each be either a link or an item."

Rather than storing or representing data in a tabular form, associative database models are depicted with data Items in nodes with Links represented by vectors or lines with arrows connecting the Items. The Link contains the relationship or association (also called vectors) between the two connected Items. Figure 6 illustrates an associative data model.

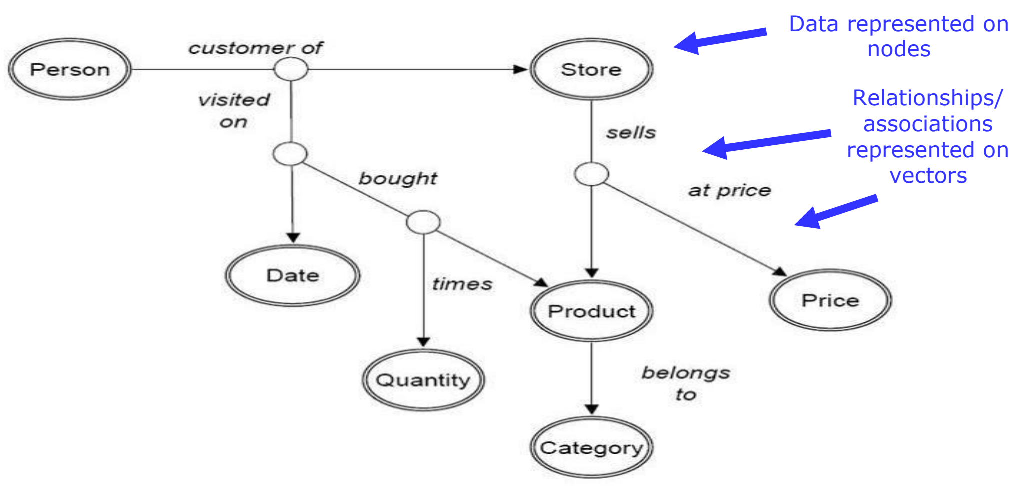

**Figure 6 Associative Data Model**

Associative database management systems (ADBMS) are not well known and do not have a large installed user base. However, two software providers have offered ADBMS products over the past 12 years. Lazysoft, based in the UK offered a product called Sentences™ until the company was shut down in 2003. Relavance Canada, Inc. offers a product called Relavance™ which has been used successfully on a variety of projects according to company literature. A number of unpublished presentations summarize the advantages of the associative database model. Muir [14] compares the two data models by noting that the "database structure that helps an RDBMS to outperform the ADBMS on the low end, hurts it on the high end … the ADBMS is also more flexible to make changes in the application, the ADBMS actually makes more sense, regardless of the size of the application." He also provides the following comparison between relational and associative database systems:

| Relational | Associative |
|---|---|
| Highly complex | Primitive structures (no complexity) |
| Resistant to change | Highly adaptable to change |
| Poor performer | High performance (no joins/no tables) |
| Limits on the types of data | No limit on data types |
| Low security | Highly secure |
| Relies on a "sequential" representation | No "sequential" dependency representation |
| Difficult system-to-system interface | Solves system-to-system interface |
| Makes COTS applications too complex | Makes COTS applications viable |
| *It's not clear which of these is worst.* | *A true breakthrough in data management.* |

Aysola [1] claims the ADBMS "is a technological solution to the problems inherent in table based data storage and management." Noting that an ADBMS works like a brain, Aysola indicates that software based on the ADBMS model can "store and find everything associatively in a Concept based, multi-dimensional information storage and management system that can be 100% compatible with the concept model of the business, yet works independently from the 'namespace' of the existing data sets (names, labels & values are attributes)." In addition, he notes that the ADBMS allows for data warehousing without tables and that data can be imported directly from existing databases into an associative database model and the original information set.

As noted earlier in this paper, software based on the associative data model has not been a major commercial success. However, one aspect of an ADBMS that may make it more attractive to future users is its inherent security features. The content (data) and index (associations) can be stored together or separately in different places. If a hacker were to access one or the other, Aysola [1] notes that "nothing more than an unlimited character string" is revealed. His presentation claims that to "make sense of one or the other you need the other component plus the algorithm used to connect the two." The separation of the data from the associations with the software as the third component creates a three part security key which enables a high level of security.

Recently, a major database company introduced a vector-based product – apparently based on the associative data model. Clark [3] reports that Ingres has developed a way for companies to rapidly search databases by creating a "query-handling portion of a database that exploits a concept known as vector processing." Benchmark data of the time to fulfill a database query was reported as follows:

- Standard Ingres database: 16.5 seconds
- Ingres VectorWise: 0.206 seconds
- Hand-written software: 0.040 seconds. [3]

Another example of applying the ADBMS model is presently underway involving data from the Human Genome Project. According to Aysola [2], the huge volume of data collected for the Human Genome Project has been imported into a Relavance™ database and is undergoing analysis. While details about the project are preliminary and cannot be published, initial results have already provided insight into the data not revealed through the use of traditional RDBMS software.

## SUMMARY

Literature about the associative database model is limited. Prior to the recent announcement by Ingres, only two software vendors have offered commercial products to support ADBMS. Muir and Aysola praise the virtues of the associative data model and the Relavance™ product specifically. The Sentences™ product is no longer offered since Lazysoft dissolved as a company. Prior to the demise of Lazysoft, King and Rainwater [9] wrote a research paper based heavily on the Sentences™ product. Their assessment of ADBMS is not greatly different than that of Muir and Aysola. King and Rainwater [9] conclude that the "associative data model is capable of being implemented as a multi-user web-enabled DBMS … and at the same time overcoming several significant limitations of the relational model." Finally, these authors contend that the associative data model "frees the information technology model to directly represent the business process: the gap between business model and database implementation has been dramatically closed" [1]. It seems likely that the future needs of users of large database systems will drive them toward database systems based on the associative data model. Subsequent research and papers should focus on quantifiable advancements and specific benefits derived from the use of ADBMS database software.

## REFERENCES

1. Aysola, K. (2009). *The impact of an associative information architecture*. Unpublished presentation.
2. Aysola, K. (2009). Personal communication, August 12, 2009.
3. Clark, D. (2009). *Corporate News: Ingres Claims Database Advance. Wall Street Journal* (Eastern Edition), p. B.6. Retrieved August 14, 2009, from ABI/INFORM Global. (Document ID: 1807665391).
4. Codd, E. F. (1970). *A relational model for large shared data banks.* New York: Association of Computing Machinery.
5. Codd, E. F. (2007). *Relational database: a practical foundation for productivity.* New York: Association of Computing Machinery.
6. Dewson, R. (2001). *Beginning sql server 2000 programming*. Birmingham, UK: Wrox Press Ltd.
7. Griffiths, M. (2001). *Introducing the associative database.* Retrieved March 30, 2009.
8. Jordan, E. W. and Machesky, J. J. (1990). *Systems development: requirements, evaluation, design, and implementation.* Boston: PWS-KENT Publishing Co.
9. King, R. S. and Rainwater, S. B. (2002). The associative data model. *Journal of Computing Sciences in Colleges.* 17(5), 154-160.
10. Kroenke, D. (1977). *Database processing: fundamentals, modeling, applications.* Chicago: Science Research Associates, Inc.
11. Kroenke, D. (2000). *Database processing: fundamentals, and Implementation* (Seventh Edition). Upper Saddle River, NJ: Prentice Hall.
12. Marston, T. (2005). *The relational data model*. Retrieved March 30, 2009.
13. McLeod, R. and Schell, G. (2001). *Management information systems* (Eighth Edition). Upper Saddle River, NJ: Prentice Hall.
14. Muir, C. (2003). *Associative technology and solutions*. Unpublished presentation.
15. Rankins, R., Bertucci, P., Gallelli, C. Silverstein, A. T. (2007). *Microsoft sql server 2005 unleashed*. Indianapolis, IN: Sams Publishing.
16. Simon, A. R. (1997). *Data warehousing for dummies*®. Hoboken, NJ: Wiley Publishing, Inc.
17. Williams, S. (2002). *The associative data model* (Second Edition), Great Britain: Lazy Software Ltd.