History Of Databases

Article *in* International Journal of Management & Information Systems (IJMIS) · December 2012

DOI: 10.19030/ijmis.v17i1.7587

Kristi Berg
Minot State University

Dr. Tom Joseph Seymour
Minot State University

History Of Databases

Kristi L. Berg, Minot State University, USA
Tom Seymour, Minot State University, USA
Richa Goel, Minot State University, USA

ABSTRACT

The database and database management systems have become an integral part of every kind of work, whether in managing business-related data or managing our household accounts. The purpose of this paper is to take a look back in time and track the origin of the database, its development over the years, and to take a look forward at what the future may hold for databases.

Keywords: Data; Database; Database Management Systems; Relational Databases

INTRODUCTION

This paper is an effort to determine the origin of databases. As we look back, we realize the history of databases can go even further back into the pre-computer era. Once the human race started to count and write and came up with a ways for measuring commodities and storing commodities, the need for compiling the information about their collectibles also came into being.

First they defined the data. The word ‘data’ is driven from the Latin word “datum” which means “to give”. It is the plural of datum, but most of the time used as a singular noun. We usually say data is available, not data are available. Datum or data means a fact, an event or a piece of information. Data is a representation of a fact, figure, and idea; it can be a number, word, or image. Thus, data refers to qualitative or quantitative attributes of a variable or set of variables. Data in itself does not really mean anything - just a cluster of numbers, words, or images out of context with no relation to the other facts. For instance, data consisting of the numbers 10 and 12 could mean anything - a date, an age of someone, an age group, or a month. However, data consisting of 10 boys and 12 girls will give more meaning to the data, showing we are counting the boys and girls in a group or set of people/class. This characterization of the data is known as the metadata - data about the data.

Once data is collected, there is a question of what to do with this abundance of data; thus, the need to organize it, store it, and make it available when needed. The human race coined the term database which is a collection of data.

HISTORY OF DATABASES

The origins of the database go back to libraries, governmental, business and medical records before the computers were invented (Date, 2003). Once people realized they needed to have the means to store data and maintain the data files for later retrieval, they were trying to find ways to store, index, and retrieve data. With the emergence of computers, the world of the database changed rapidly, making it an easy, cost effective, and less space-consuming task to collect and maintain the database.

Database Definition

A database is a self-describing collection of integrated records. A record is a representation of some physical or conceptual object. A database is self-describing in that it contains a description of its own structure. This description is called metadata - data about the data. The database is integrated in that it includes the relationships among data items, as well as including the data items themselves (Kroenke & Auer, 2007).

THE EVOLUTION OF THE COMPUTERIZED DATABASE

Ancient to Modern

Humans began to store information long ago. In ancient times, elaborate database systems were developed by government offices, libraries, hospitals, and business organizations, and some of the basic concepts and principles used in building those systems are still being used today. From ancient times to relational and object-relational systems, database technology has gone through several generations and its history and progress through the times is fascinating.

1960's

With computers gaining popularity and becoming cost effective to use by private companies to increase their data storage capacity, the 60's and mid-60's can be considered as the new direction in the field of the database (DuCharme, 2012). The introduction of the term database coincided with the availability of direct-access storage (disks and drums) from the mid-1960s onward. The term represented a contrast with the tape-based systems of the past, allowing shared interactive use rather than daily batch processing. Two main data models were developed - network model CODASYL (Conference on Data System Language) and hierarchical model IMS (Information Management System).

The first generation of database systems was navigational. Applications typically accessed data by following pointers from one record to another. Storage details depended on the type of data to be stored. Thus, adding an extra field to database required rewriting the underlying access/modification scheme. Emphasis was on records to be processed, not overall structure of the system. A user would need to know the physical structure of the database in order to query for information. One database that proved to be a commercial success was the SABRE system that was used by IBM to help American Airlines manage its reservations data (Bercich, 2002). This system is still utilized by the major travel services for their reservation systems.

1970's

The relational database model was conceived by E.F. Codd in 1970. This model was a departure from tradition by insisting that applications should search for data by content rather than by following links. His system can be defined using two terminologies (Taylor, 2000). Instance, a table with rows and columns and schema, specifies the structure, including the name of relation, name, and type of each column.

Codd's model was based on branches of mathematics called set theory and predicate logic. His ideas changed the way people thought about databases. In his model, the database's schema, or logical organization, is disconnected from physical information storage, and this became the standard principle for database systems. During the period 1974 to 1977, two major relational database system prototypes were created:

1. INGRES was developed at the University of California-Berkeley and became commercial and followed up POSTGRES which was incorporated into Informix. This ultimately led to Ingres Corp., Sybase, MS SQL Server, Britton-Lee, and Wang's PACE. This system used QUEL as query language.
2. System R was developed at IBM in San Jose and led IBM's SQL/DS & DB2, ORACLE, HP's Allbase, and Tandem's Non-Stop SQL. DB2 became one of the first DBMS (Database Management System) product based on the relational model (Oppel, 2011).

A new database model called Entity-Relationship, or ER, was proposed by P. Chen in 1976. This model made it possible for designers to focus on data application instead of logical table structure (Taylor, 2000). The term Relational Database Management System (RDBMS) was coined during this period.

Table 1: Relational DBMS at a Glance (Mullins, 2002)

Fundamental Relational Database Characteristics	**Database Schema**	**DBMS Functions**	**Database Approach**	**Advantages and Disadvantages of DBMSs**
The internal structure of an operating database is basically fixed in the 'row' direction. The user will interact with a logical view of the data and need not know anything about the actual internal structure.	1. Conceptual Schema logically describes all data in the database. 2. Internal Schema (Physical Schema) describes how data are actually stored. 3. External Schema (User View) describes the data that is of use to the user.	1. Data dictionary management 2. Data Storage Management 3. Data Transformation and presentation 4. Security Management 5. Multi-user access control 6. Backup and recovery management 7. Data integrity management 8. Database language and application programming interfaces 9. Database communication interfaces	1. *Data Definition Language (DDL):* define database structure (e.g. tables) 2. *Data Manipulation Language (DML):* to retrieve, insert, delete and update data in the database; Query language are part of DML. 3. *Data Control Language (DCL):* control the access of data.	*Advantages:* 1. Control of data redundancy, consistency, abstraction, sharing 2. Improved data integrity, security, enforcement of standards and economy of scale 3. Balanced conflicting requirements 4. Improved data accessibility, responsiveness, maintenance 5. Increased productivity, concurrency, backup and recovery services *Disadvantages:* 1. Complexity, size, cost of DBMS's 2. Higher impact of a failure

Starting in the 1960's with the network model to the beginning of the 1980's, the database experienced vigorous changes. Three models for organizing the data were already developed - the network model, the hierarchic model, and the relational model. Once the relational model became prevalent, the history of databases took a sharp turn and started heading toward a better future for data management - data storage and data integrity.

Table 2: Outline of the Concepts Behind The Three Models - Network, Hierarchic and Relational (Petersen, 2002)

Concepts	**Relational**	**Network**	**Hierarchic**
Item	Role name/domain	Data item type	Item/field
Item Value	Component	Data item occurrence	Value
Group	Not allowed	Group	Group
Entity (type)	Relation	Record Type	Entry/segment type
Entity Instance	Tuple	Record occurrence	Entry/segment occurrence
Relationship	Foreign key comparable underlying domains	Set type	Hierarchic (implied)
Relationship Instance		Set occurrence	Assembly
Data administrator view	Data model	Logical structure	Logical structure
Definition of data administrator view	Data model definition	Schema	Schema
User View	Data sub model		
Definition of user view	Data sub model definition	Sub schema	Sun schema
Data-base subdivision		Area	
Entry Points	Primary key	Singular sets CALC records	Root group
Single unique/identifier	Candidate key	Key	Root segment sequencer (unique)

1980's

Commercialization of relational systems begins as a boom in computer purchasing fuels the DB (database) market for business. SQL (Structured Query Language) became the intergalactic standard (Taylor, 2007). DB2 became IBM's major product. Network and hierarchical models lost their charm with no further development of these systems, but some legacy systems are still in use today. Development of IBM's PC gave rise to many DB companies and products, such as RIM, RBASE 5000, PARADOX, OS/2 Database Manager, Dbase III, IV (later FoxBASE, and Visual FoxPRO), and Watcom SQL. All these systems were introduced during the1980's and were based on the relational model.

During this period, the concept of the object-oriented database was developed. The term 'object-oriented database system' first appeared around 1985. An object database, also object-oriented database management system, is one in which information is represented in the form of objects as used in object-oriented programming (Haadi, 2010). Object databases are different from relational databases and belong together to the broader database management system. Object database's main usage is in object-oriented areas. Some object-oriented databases are designed to work well with object-oriented programming languages, such as Delphi, Ruby, Python, Perl, Java, C#, and Visual Basic .NET, C++, Objective-C, and Smalltalk. Others have their own programming languages. OODBMS (Object-Oriented Database Management System) uses exactly the same model as object-oriented programming languages. OODBMS supports the modeling and creation of the data as objects.

OODBMS could efficiently manage a large number of different data types. Objects with complex behaviors were easy to handle using inheritance and polymorphism. This also helped in reducing the large number of relations by creating objects. The biggest problem with OODBMS was switching an existing database to OODBMS, as the transition requires an entire change from scratch and it is typically tied to a specific programming language and an API (Application Programming Interface) which reduces the flexibility of the database. To overcome the problems of OODBMS and take full advantage of the relational model and object-oriented model, the Object Relational Database Model was developed in the early 1990's.

1990's

In the early 1990's, an industry shakeout resulted in fewer surviving companies offering increasingly complex products at higher prices. Much development during this period centered on client tools for application development, such as PowerBuilder (Sybase), Oracle Developer, and VB (Microsoft). The client-server model for computing became the norm for future business decisions (DuCharme, 2012). This time period also marked the development of personal productivity tools, such as Excel/Access (MS) and ODBC (Open Database Connectivity).

Around the mid-1990's, the usable Internet/World Wide Web appeared. This allowed remote access to computer systems with legacy data. Client-server frenzy reached the desktop of average users with little patience for complexity while Web/DB grew exponentially.

The large investment in Internet companies fueled the tools market boom for Web/Internet/DB connectors. Active Server Pages (ASP), Front Page, Java Servlets, JDBC, Enterprise Java Beans, ColdFusion, Dream Weaver, and Oracle Developer 2000 were examples of such offerings. Open source solutions came online with widespread use of GCC (GNU Complier Collection), CGI (Computer Generated Imagery), Apache, and MySQL. Online Transaction Processing (OLTP) and Online Analytic Processing (OLAP) came of age with many merchants using point-of-sale (POS) technology on a daily basis (DuCharme, 2012).

As mentioned earlier, to overcome the problems of OODBMS and fully explore the possibilities of relational model, ORDBMS (Object Relational Database Management System) also emerged during the 1990's. The basic goal for the object-relational database was to bridge the gap between relational databases and the object-oriented modeling techniques used in programming languages, such as Java, C++, Visual Basic .NET or C#. However, a more popular alternative for achieving such a bridge is to use standard relational database systems with some form of object-relational mapping (ORM) software (Oppel, 2011). Whereas traditional RDBMS or SQL-DBMS products focused on the efficient management of data drawn from a limited set of data types (defined by the relevant language standards), an object-relational DBMS allowed software developers to integrate their own types and the methods that apply to them into the DBMS.

Another major development which took place during 1997 was the introduction of the Extensible Markup Language (XML) (DuCharme, 2012). XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C - and several other related specifications - all gratis open standards. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, such as web services. Many application programming interfaces (APIs) have been developed for software developers to use to process XML data, and several XML schema systems exist to aid in the definition of XML-based languages. The XML applied to database processing which solved long-standing database problems. Major vendors begin to integrate XML into DBMS products.

Early 21st Century

The year 2000 - the fear of the Y2K - was the main reason for the fast pace technology changes which took place in the late 1990's. Once the 2000's arrived, there was a steep decline in the Internet industry as a whole, but solid growth of database applications continued. More interactive applications appeared with the use of PDAs (Personal Data Assistants), Point-of-sales transactions, and consolidation of vendors. Three main companies dominated the larger market share of databases - IBM, Microsoft, and Oracle.

Databases do a good job of storing data and recounting data but still often rely on human interaction or an external program to make correlations and decisions based on the data. The step in this direction started in the early 21st century and is still in its infancy. There is a growing trend now to provide more sophisticated programming logic within the database structure (Databases, 2003). Initially, databases only controlled what kind of data could be put in a particular field. For example, if the defined field is for entering the date\time, it would trigger an error message if some other value is inserted into the field. Databases have become more sophisticated and have features such as triggers, cascading update, and delete. This prevented updaters from creating inconsistencies between the tables. Databases also developed a simplified procedural language that contained embedded SQL with some looping and control structures; for example, Sybase/SQL Servers' transact SQL, Oracle's PL/SQL, and PostgresSQL.

Huge terabyte systems are also appearing and will require novel means of handling and analyzing data, such as large science databases like the genome project, the geological, national security, and space exploration data. Data mining, data warehousing, and data marts are commonly used techniques today.

NoSQL

Carlo Strozzi used the term NoSQL in 1998 to name his lightweight, open-source relational database that did not expose the standard SQL interface. Eric Evans, a Rackspace employee, reintroduced the term NoSQL when Johan Oskarsson of Last.fm wanted to organize an event to discuss open-source distributed databases (Taylor, 2007). The name attempted to label the emergence of a growing number of non-relational, distributed data stores that often did not attempt to provide ACID (Atomicity, Consistency, Isolation, Durability) guarantees, which are the key attributes of classic relational database systems, such as Sybase, IBM DB2, MySQL, Microsoft SQL Server, Postgres SQL, Oracle RDBMS, Informix, and Oracle RDB.

Often, NoSQL databases are categorized according to the way they store the data and they fall under categories such as key-value stores, BigTable implementations, document store databases, and graph databases. NoSQL database systems rose alongside major internet companies, such as Google, Amazon, Twitter, and Facebook, which had significantly different challenges in dealing with data that the traditional RDBMS solutions could not cope. NoSQL databases are often highly optimized for retrieve and append operations and often offer little functionality beyond record storage. The reduced run time flexibility, compared to full SQL systems, is compensated by significant gains in scalability and performance for certain data models.

UnQL

In 2011, DB developers started working on UnQL (Unstructured Query Language), a specification for a query language for NoSQL databases. It is built to query collections of documents with loosely defined fields versus tables with rows and columns (Oppel, 2011). UnQL is claimed to be a superset of SQL within which SQL is a very constrained type of UnQL for which the queries always return the same fields, same number, names and types. However, UnQL does not cover the data definition language (DDL) SQL statements like CREATE TABLE or CREATE INDEX.

FUTURE TRENDS IN DATABASES

This paper has taken a look back at the history of databases to establish their importance. Now is the time to look forward to the future and to try to determine "what the future holds for databases".

Currently databases are beginning to take on even more complex logic. Another feature that is being expanded is the concept of separation of location from the abstract concept of the database itself; which Codd defined long ago. This feature enables a database to reside in more than one location and to be queried as a continuous unit. Such instances are called distributed or federated databases. A portion of a database can be in New York and another in Boston and a query requested to count all the customers would then be run simultaneously at both locations. This is also made possible due to the increase in the speed of networks. Arnold (2008) predicts the following trends in the world of databases.

- Database and data management are at a turning point.
- Among the drivers are changes in architecture, like cloud computing and the need to deal with large amounts of data.
- Databases will be used outside traditional boundaries. One example is Google's MapReduce. Data collections - not databases - are increasingly important for capturing and linking knowledge. The cloud, mobile, and virtual applications are game changers.

With time, the storing of data is becoming more and more complex. Spatial data requires the need for specialized functions. Image data, scanned-in data, and complex medical data, such as gene sequences, also need to be stored. Medical devices that record physical data and transform it into a digital format are in use and need to be processed by computers. The processing of this data to derive patterns, such as the DNA sequences that will cause a particular disease, also demands more processing and data storage capabilities.

CONCLUSION

The future of databases appears bright with possibilities. Huge systems are appearing and will require novel means of handling and analyzing data. Mobile database use is a product now already in the market. Ever-changing and improving technology has made people expect all information to be only a click or call away. Distributed transaction processing is becoming the norm for business planning in many arenas.

Perhaps the day is not far too away when the databases will become advanced enough to store digital replicas of ourselves. The idea of the Star Trek transporter transporting a person from one location to another may then become a simple matter of radio waves transporting digital data that defines the essence of what we are to a distant planet. The future of databases looks to be very promising.

AUTHOR INFORMATION

Kristi Berg is an assistant professor of Management Information Systems at Minot State University in Minot, North Dakota. She received her Bachelors in Business Education and Master of Science in Management with an Information Systems Concentration from Minot State University, and her Ph.D. in Organization Management with an Information Systems Specialization from Capella University, Minneapolis, Minnesota. Her wide area of interest

includes management information systems, database theory and application, virtual business, and workforce development. E-mail: kristi.berg@minotstateu.edu

Dr. Tom Seymour was appointed to Chair the Business Information Technology Department for 2007-2009. Dr. Seymour graduated from Mayville (BS), UND (MA), and Colorado State University (PhD). He came to Minot State University from Murray, Kentucky after teaching in 7 states. He has given over 150 Computer / E-Commerce presentations in 41 states and 10 foreign countries. Dr. Seymour teaches technology classes in the classroom and via the Internet. Tom is a HLC/ NCA peer reviewer and has reviewed in 19 states including Singapore, Mexico and China. His publication record includes publishing over 80 articles in refereed journals and editing many proceedings and college textbooks. For five years Tom wrote an Internet column for the *Minot Daily News* and Dr. Seymour is Past-President of the International Association for Computer Information Systems and the 2011 IACIS Ben Bauman Award for Excellence Awardee. E-mail: tom.seymour@minotstateu.edu (Corresponding author)

Richa Goel is a graduate student in the Masters of Science in Information Systems at Minot State University, Minot, North Dakota. E-mail: richa.goel@my.minotstateu.edu

REFERENCES

1. Arnold, S. (2008). The Future of Database. Retrieved from http://arnoldit.com/wordpress/2008/10/21/the-future-of-database/ (2012, February 18).
2. Bercich, N. (2002). “The Evolution of the Computerized Database”, *Concept, 26*. Retrieved from http://concept.journals.villanova.edu/article/view/311/274
3. Databases: past, present, and future. (2003). Retrieved February 19, 2012, from http://www.paragoncorporation.com/ArticleDetail.aspx?ArticleID=20
4. Date, C. J. (2003). *An Introduction to Database Systems, Fifth Edition*. Boston, MA: Addison Wesley.
5. DuCharme, B. (2012, January 26). 25 years of database history (starting in 1955). [Web blog post]. Retrieved from http://www.snee.com/bobdc.blog/
6. Haadi, M. (2010, October 18). All about Database. [Web blog post]. Retrieved from http://mhaadi.wordpress.com/2010/10/18/the-evolution-of-database/
7. Kroenke, D. & Auer, D. (2007). *Database Concepts.* 3rd ed. New York, NY: Prentice.
8. Mullins, C. (2002). *Database Administration: The complete guide to practice and procedures*. Indianapolis, IN: Addison-Wesley Professionals.
9. Oppel, A. (2011). *Databases DeMYSTiFieD—Hard Stuff made easy*. New York, NY: McGraw-Hill.
10. Petersen, J. (2002). *Absolute Beginner's Guide to Databases.* Indianapolis, ID: Que.
11. Taylor, A. (2000). *Database Development for Dummies*. New York, NY: John Wiley & Sons, Inc.
12. Taylor, A. (2007): *SQL for Dummies.* New York, NY: John Wiley & Sons, Inc.

NOTES