2. Data storage structure in Database Management System.

2.1 Data storage structures in pre-digital times

In ancient times, people already had the awareness to record data. As early as the Late Stone Age (50,000 to 12,000 years ago), animal bones for producing tally sticks, as exemplified by Ishango Bone, demonstrate that people used sticks or bones with notches marked on them for trade or supply activities (Heidari et al., 2022). This method of recording is based on the principle of split counting, taking advantage of the fact that the same stick or bone breaks a compatible interface and records the same data at both ends, thus ensuring that the transaction record is not altered by one side unilaterally. This structure is similar to two associated fields in a modern relational database.

Incan Quipu is a collection of coloured knotted cotton threads used as data storage equipment by the ancient Incas in 3000 B.C. It has one main rope that would have a variety of different coloured sub-ropes on it. Each colour of string had a different meaning, such as the number of men in the village, the number of pregnant women, the amount of food, etc. The ancient Incas tied knots in these different coloured sub-ropes to record data (Keister, 1964). This is similar to a table in a modern database, where the main rope can be a table that stores information about a village, the different coloured sub-ropes are fields in it, and the number of knots is a value of type int.

With the development of human civilisation, the scale of data that needs to be stored has grown, and with the advent of paper, paper-based data management systems have emerged as the longest-running form of record keeping until nowadays. (McCarthy, 2022) Take the example of the Great Library of Alexandria, whose first director, Zenodotus, divided the collection into three main categories and ordered them by author's name, which was the first recorded use of metadata. (Phillips, 2010) From here, the concept of a database management system began to emerge. The whole library as a database, and each category of books as a table, in which the books as fields and data are organized in a certain order by the librarian (database operator).

2.2 Modern data storage structures

After being first described by E.F. Codd in 1970, and after half a century of development, database-related concepts have matured considerably. Modern database management systems are divided into two main categories according to their storage structure: relational databases and non-relational databases.

2.2.1 Relational databases

The relational database was first proposed by E.F. Codd in 1970 (Codd, 1970), the main feature of the relational database is the consistency of affairs. It emphasises the "ACID rules", namely Atomicity, Consistency, Isolation and Durability. Relational databases store data in a structured way, with data types defined before they are deposited and complex relationships between tables. The language is mainly SQL and the representative database management system is MySQL.

The logical structure of the relational model of a relational database is a two-dimensional data structure, which uses this simple logical structure to store the data of entities and describe the relationships between them (Haiyan et al., 2010). Relational databases generally store a structure of b+ -tree (Kieseberg et al., 2011), which is a structure with a relatively stable time complexity for both insertion and modification.

2.2.2 Non-relational databases

Unlike relational databases, non-relational databases do not use a relational model and are refined to support data types and query methods (EdPrice-MSFT, n.d.). Due to the dramatic increase in the volume of modern data, non-relational databases have an irreplaceable advantage over traditional relational databases when faced with large quantities of data. (Nishtha, 2012)

a) Key-value Database Model

The key-value data storage model is used by many enterprises to store data, such as Facebook's five Memcached use cases which receive millions of requests per day. (Atikoglu et al., 2012) It is a large Hashmap that stores data through the key and value association structure. Its most important feature is its high concurrency and performance. For example, SILT can reach 46k queries or 23k inserts per second (Lim et al., 2011). The represented database management systems are Memcached and Redis.

b) The graphical database model

The graphical database model is a storage model that summarises the data and relationships of entities in diagrams, which consist of three parts: data structure, transformation language and holistic constraints (Angles & Gutierrez, 2008). It is mainly used in situations where the relationships are more important than the data itself. Its advantage is that all information about an entity can be explicitly and clearly represented in association with other nodes (Paredaens et al., 1995).

c) Network database model

The network database model allows tree-structured children to have multiple pairs of parents, solving the hierarchical structure of many-to-many relationships (MongoDB, n.d.) The network database model is simple to design and efficient to access (Raima, n.d.). This structure is well-suited for quick searches, data access or navigation. A representative database management system is Raima.

d) The Document Database Model

A document database is a database that uses documents as a data storage structure, a database type that considers documents as part of the database (Clifton & Garcie-Molina, 1988). As the most similar non-relational database model to the relational database, the document database model is intuitive, flexible and has a variety of APIs that allow it to be adapted to a wide range of development options (MongoDB, n.d.). JSON and MongoDB are representatives of the document database model.

2.3 Similarities and differences between pre-digital storage structures and modern storage structures

2.3.1 Similarities

Both pre-digital and modern database storage structures have certain correspondences between data, including one-to-many, one-to-one, many-to-many, etc. They also have a simple reflection of the ACID principle, such as a certain Consistency and Durability of the bones of a record trade record. Secondly, the pre-digitised database storage structure possesses a relatively simple storage structure and logical structure, such as the classification of books, sorting by author name, etc.

2.3.2 Differences.

a) Stronger management of data integration and cleansing

Pre-digital databases did not store logically linked information as tightly as modern databases, which led to inefficiencies in integration and cleansing. Pre-digital database management systems were generally paper-based, so data was managed manually. When integrating data, it was necessary to manually locate the paper-based information that needed to be used, bring it together and even transcribe it. Similarly, when cleaning data, it is necessary to manually check the data and eliminate errors manually. Modern databases, on the other hand, allow for uniform and rapid batch manipulation of data by methods internal or external to the database management system. As mentioned by Nin, manual operations are quickly integrated with the aid of algorithms to clean the data(Nin et al., 2007). Modern database management systems are considerably more efficient compared to pre-digitalisation.

b) Greater logic

Modern database management systems also have the advantage of greater logical expressiveness. Thanks to the development of computers, it has become possible to optimise storage methods in terms of structure. Before digitisation, it was difficult to represent tree hierarchies or complex entity relationships, and the description methods were more homogeneous, mostly sorted or ordered according to certain simple primary keys. In modern databases, on the other hand, we have different data structures to target our different situations. For example, the relational database model has foreign keys, the graph database model emphasises relationships more than the data itself, the file database model has a great deal of flexibility that was not available before digitisation, etc. Well-tested and validated data structures can be easily used by anyone and are more adapted to the processing of big data.

c) Lower redundancy and fewer errors improve data consistency

The adoption of a modern database structure has the added benefit of lower redundancy compared to pre-digitisation databases. Pre-digital databases had no logical structural constraints and the same information could be stored repeatedly. After a particular piece of information has been updated, there may also be information that has been forgotten to be updated leading to inconsistencies in the data. In modern databases, however, inconsistency errors can be substantially reduced through constraints and redundancy through database design paradigms, guided by database design principles, which are made possible by digitised storage and enhanced logical relationships.

d) Higher hit-rate searches

Searching with a modern database structure has a higher hit rate than with a pre-digital database structure. Before digitisation, people searched the contents of databases mainly manually and sequentially, requiring a traversal of a category of books. In modern databases, the storage structure may be a b+ tree (Kieseberg et al., 2011) and the logical structure may also refer to precise targeting as in the document example designed by Chris Clifton and others (Clifton & Garcie-Molina, 1988). This can make information retrieval much more efficient and reduce search times.

e) High reusability for analysis and exploitation

Modern databases tend to be more reusable than pre-digital databases due to their well-organised data structure. When analysing different aspects of information, one can quickly obtain results from a modern database with relatively good relationships and data, whereas pre-digital databases often take a lot of time to analyse due to the relative weakness of the relationships.

Reference

Heidari, A., Jafari Navimipour, N. and Unal, M. (2022) “The history of computing in Iran (persia)—since the achaemenid empire,” Technologies, 10(4), p. 94. Available at: <https://doi.org/10.3390/technologies10040094>.

Keister, O. R. (1964). The Incan Quipu. The Accounting Review, 39(2), 414–416. Available at: <http://www.jstor.org/stable/243533>

McCarthy, G. (2022) The past, present and the future of Data Management: GSM barcoding, Barcoding. Available at: https://www.barcoding.co.uk/the-past-present-and-the-future-of-data-management/ (Accessed: November 13, 2022).

Phillips, H. (2010). Great Library of Alexandria. Library Philosophy and Practice, Aug.. Available at: <https://nbn-resolving.org/urn:nbn:de:0168-ssoar-190807>

Codd, E.F. (1970) “A relational model of data for large shared data banks,” Communications of the ACM, 13(6), pp. 377–387. Available at: <https://doi.org/10.1145/362384.362685>.

Haiyan, Y. et al. (2010) “Performance evaluation of post-relational database in hospital information systems,” 2010 Second International Workshop on Education Technology and Computer Science [Preprint]. Available at: <https://doi.org/10.1109/etcs.2010.134>.

Kieseberg, P. et al. (2011) “Using the structure of b+-trees for enhancing logging mechanisms of databases,” Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services - iiWAS '11 [Preprint]. Available at: <https://doi.org/10.1145/2095536.2095588>.

EdPrice-MSFT (no date) Non-relational data and NoSQL - Azure Architecture Center, Azure Architecture Center | Microsoft Learn. Available at: https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data (Accessed: November 13, 2022).

Nishtha, J. et al. (2012) “A Survey and Comparison of Relational and Non-Relational Database”, International Journal of Engineering Research & Technology (IJERT), 1(6). Available at: <https://www.academia.edu/44482078/IJERT_A_Survey_and_Comparison_of_Relational_and_Non_Relational_Database>.

Atikoglu, B. et al. (2012) “Workload Analysis of a large-scale key-value store,” Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems - SIGMETRICS '12 [Preprint]. Available at: <https://doi.org/10.1145/2254756.2254766>.

Lim, H. et al. (2011) “Silt,” Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles - SOSP '11 [Preprint]. Available at: <https://doi.org/10.1145/2043556.2043558>.

Angles, R. and Gutierrez, C. (2008) “Survey of graph database models,” ACM Computing Surveys, 40(1), pp. 1–39. Available at: <https://doi.org/10.1145/1322432.1322433>.

Paredaens, J., Peelman, P. and Tanca, L. (1995) “G-log: A graph-based query language,” IEEE Transactions on Knowledge and Data Engineering, 7(3), pp. 436–453. Available at: <https://doi.org/10.1109/69.390249>.

MongoDB (no date) Understanding the network database model, MariaDB KnowledgeBase. Available at: https://mariadb.com/kb/en/understanding-the-network-database-model/ (Accessed: November 13, 2022).

Raima (no date) Network database, relational DB, and graph DB compared, Raima. Available at: https://raima.com/network-database-relational-db-and-graph-db-compared/ (Accessed: November 13, 2022).

Clifton, C. and Garcie-Molina, H. (1988) “The design of a document database,” Proceedings of the ACM conference on Document processing systems - DOCPROCS '88 [Preprint]. Available at: <https://doi.org/10.1145/62506.62528>.

MongoDB (no date) Document database - nosql, MongoDB. Available at: https://www.mongodb.com/document-databases (Accessed: November 13, 2022).

Nin, J. et al. (2007) “On the use of semantic blocking techniques for data cleansing and integration,” 11th International Database Engineering and Applications Symposium (IDEAS 2007) [Preprint]. Available at: https://doi.org/10.1109/ideas.2007.4318104.