INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

SPECIAL ISSUE FOR
INTERNATIONAL CONFERENCE ON
"INNOVATIONS IN SCIENCE & TECHNOLOGY: OPPORTUNITIES & CHALLENGES"

EVOLUTION OF DATABASE MANAGEMENT SYSTEM AND ITS IMPACT ON GROWTH OF INDUSTRIES

MS. SHRADDHA S. SURATKAR

Dept. of Computer Engineering, VJTI, Matunga

Accepted Date: 07/09/2016; Published Date: 24/09/2016

Abstract: *Databases have been known from a long time. In simpler words it is collection of tables related to each other which allows retrieval of data. But today the world has moved from this naive definition of database. After going through several generations, the history of database technology has influenced over more than 40 years now. In this age of Big Data, everyone with a computer system seems to want to persist, query and manipulate data. The data in databases is typically organized to mold aspects of reality in a way that supports and processes requiring information. Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software's that allows users to interact with one or more databases. Well-known DBMSs include Oracle, Sybase, IBM DB2, MySQL, PostgreSQL, Microsoft SQL Server . The next generation of post-relational databases in the late 2000s became known as NoSQL database introducing fast key values stores and document-oriented databases. A competing "next generation" known as NewSQL databases is targeting to match high performance of NoSQL database. Thus, the evolution of databases right from navigational one to NewSQL is a significant achievement. This paper will help in understanding the gradual evolution of databases management system and its impact in the growth of industries.*

Keywords: Database management system, RDBMS, NoSQL, NewSQL

PAPER-QR CODE

Corresponding Author: MS. SHRADDHA S. SURATKAR

Co Author:

Access Online On:

www.ijpret.com

How to Cite This Article:

Shraddha S. Suratkar, IJPRET, 2016; Volume 5 (2): 309-318

INTRODUCTION

In the last few years, we've seen the rise of a new breed of DBMS Models. From pre-stage flat-file system, to relational and object-relational systems, there have been several generations in database technology. As data is a vital organisational resource, it needs to be managed like other important business assets. Most of the internal and external operations of industries could not survive without quality data. Thus, DBMs is a specialized computer software responsible for managing the data [1].The goals of DBMS are:

i. Allow users to create new databases and specify their schemas (logical structure of data).
ii. Give users the ability of modifying and querying the data.
iii. Support the storage of very large amounts of data, allowing efficient access to data for queries and database modifications.
iv. It monitors performance, maintain data integrity, dealing with concurrency control, and recovers information that has been corrupted by some event such as an unexpected system failure.
v. Control access to data from many users at once in isolation and ensure the actions on data to be performed completely.
vi. Enable durability, the recovery of the database in the face of failures.
vii. Provides logging and auditing the activities.
viii. Responsible for automated roll backs.

The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible[4]. The DBMS can offer both physical and logical data independence. That means users and applications are prevented from needing to know where data is stored or having to be concerned about changes to the physical structure of data i.e. storage and hardware. As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because few changes have been made to the database [2].

HISTORY OF DATABASE MANAGEMENT SYSTEM

Evolution of DBMS

From Flat File System to relational and object-relational system, DBMS has continued to do heavy lifting in data management [3]. They play prominent roles in addressing the new enterprise requirements. Figure below (Figure 1) shows all popular DBMS in a time line. Timeline varies from 1980's till current date.

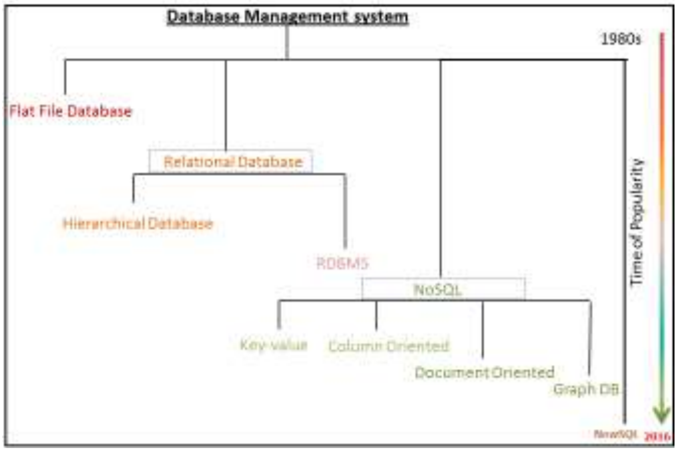
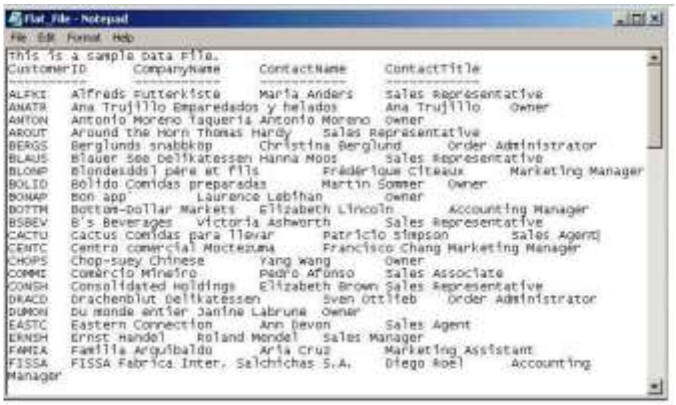


Figure 1: Growth of DBMS

1. Navigational databases/ Flat File Databases

Data was maintained in a punched cards which were later replaced by flat file. Very primitive analytics were possible on these database. Sequential access method and random access methods were used to access the database but this DBMS were not strongly secure. The problem of data duplication occurred as same data was hold by different programs wasting space and resources (Figure 2). It had high maintenance costs for ensuring data consistency and controlling access.



```
This is a sample Data File.
CustomerID      CompanyName      ContactName      ContactTitle
----------      -----------      -----------      ------------
ALFKI   Alfreds Futterkiste      Maria Anders     Sales Representative
ANATR   Ana Trujillo Emparedados y helados        Ana Trujillo    Owner
ANTON   Antonio Moreno Taqueria Antonio Moreno  Owner
AROUT   Around the Horn Thomas Hardy    Sales Representative
BERGS   Berglunds snabbkop       Christina Berglund      Order Administrator
BLAUS   Blauer See Delikatessen Hanna Moos        Sales Representative
BLONP   Blondesddsl père et fils        Frédérique Citeaux      Marketing Manager
BOLID   Bólido Comidas preparadas       Martin Sommer   Owner
BONAP   Bon app         Laurence Lebihan         Owner
BOTTM   Bottom-Dollar Markets   Elizabeth Lincoln       Accounting Manager
BSBEV   B's Beverages   Victoria Ashworth        Sales Representative
CACTU   Cactus Comidas para llevar      Patricio Simpson        Sales Agent
CENTC   Centro comercial Moctezuma      Francisco Chang Marketing Manager
CHOPS   Chop-suey Chinese        Yang Wang        Owner
COMMI   Comércio Mineiro         Pedro Afonso     Sales Associate
CONSH   Consolidated Holdings   Elizabeth Brown Sales Representative
DRACD   Drachenblut Delikatessen        Sven Ottlieb    Order Administrator
DUMON   Du monde entier Janine Labrune  Owner
EASTC   Eastern Connection       Ann Devon        Sales Agent
ERNSH   Ernst Handel    Roland Mendel   Sales Manager
FAMIA   Familia Arquibaldo       Aria Cruz        Marketing Assistant
FISSA   FISSA Fabrica Inter. Salchichas S.A.    Diego Roel      Accounting
Manager
```

Figure. 2 Flat File database

2. Relational Database (Hierarchical databases):

In this model, files are related in a parent/child manner, with each child file having at most one parent file. These databases however can solve many purposes, its applications are restricted to one-to-one mapping data structures (Figure 3). For example, it will work well if this data structure has to show job profile hierarchy in a corporate. But the structure will fail if the reporting becomes slightly more complicated and a single employee reports to many managers. Hence, it was thought to have a database structures which can have different kinds of relations. This type of structure should allow one-to-many mapping. Such table came to be known as Relational database management system (RDBMS) (Figure 4).

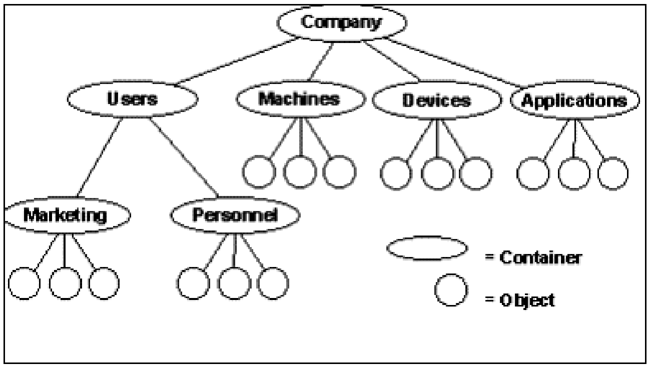
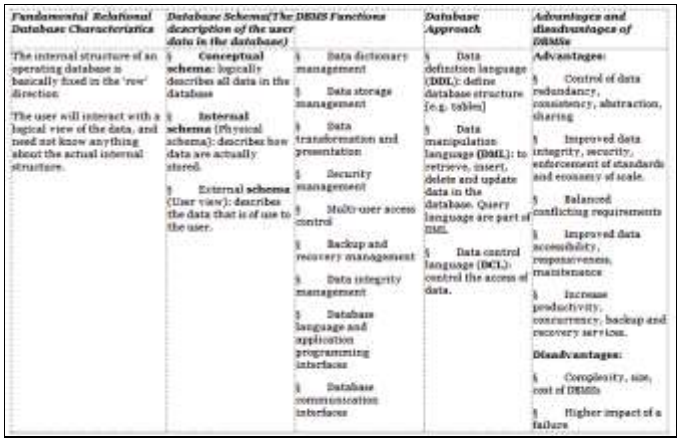


Figure 3: Hierarchical databases

This kind of data storage optimizes disc space occupied without compromising on data details. This is the data base which is generally used by the analytics industry. However, when the data looses a structure, such data base will be of no help. Also, due to lack of standards, it can't easily handle many-many relationships.



Fundamental Relational Database Characteristics	*Database Schema(The description of the user data in the database)*	*The 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.	§ **Conceptual schema**: logically describes all data in the database § **Internal schema** (Physical schema): describes how data are actually stored. § External **schema** (User view): describes the data that is of use to the user.	§ Data dictionary management § Data storage management § Data transformation and presentation § Security management § Multi-user access control § Backup and recovery management § Data integrity management § Database language and application programming interfaces § Database communication interfaces	§ Data definition language (**DDL**): define database structure [e.g. tables] § Data manipulation language (**DML**): to retrieve, insert, delete and update data in the database. Query language are part of DML. § Data control language (**DCL**): control the access of data.	**Advantages:** § Control of data redundancy, consistency, abstraction, sharing § Improved data integrity, security, enforcement of standards and economy of scale. § Balanced conflicting requirements § Improved data accessibility, responsiveness, maintenance § Increase productivity, concurrency, backup and recovery services. **Disadvantages:** § Complexity, size, cost of DBMSs § Higher impact of a failure

Figure 4: Relational Databases

3. Object Oriented Database Model:

It supports modeling and creation of data as objects. It Can efficiently manage a large number of different data types. Objects with complex behaviors are easy to handle using inheritance and polymorphism etc. Also, it reduces the large number of relations by creating objects (Figure 5). But this database is typically tied to a specific programming language and an API; this reduces its flexibility. Ad-hoc queries are difficult to implement as one cannot join two classes as one can join two tables in RDBMS. Therefore, queries depend upon the design of the system. Also, it creates problems when deleting data in bulk.

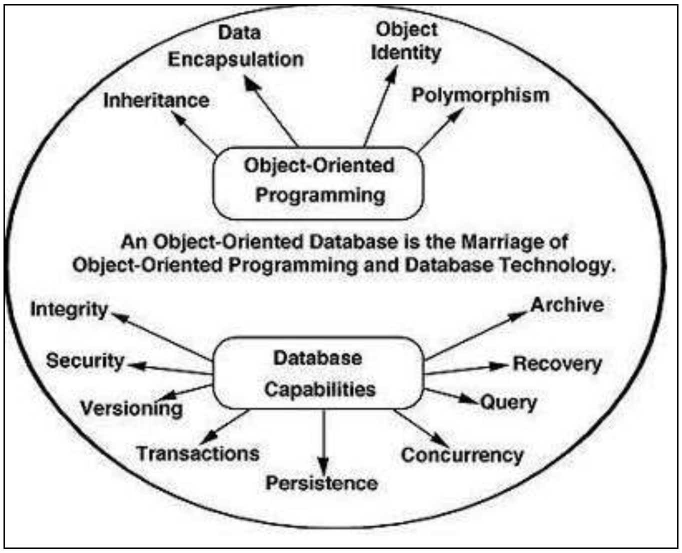


Figure 5: Object Oriented Databases

4. NoSQL:

NoSQL is often known as "Not Only SQL". When it was difficult tonnes of data to mine using RDBMS, they started exploring ways to store such datasets. Anything which is not RDBMS today is loosely known as NoSQL. After social networks gained importance in the market, such database became common in the industry. Following is an example where it will become very difficult to store the data on RDBMS (Figure 6) :

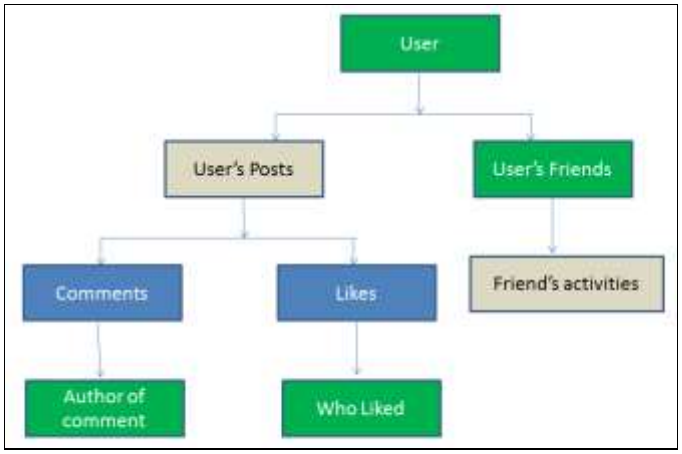


Figure 6: Structure of Data in NoSQL

If the entire data from social networking site like facebook, twitter is to be stored in RDBMS then for executing a single query which can be just a response of opening home page, multiple tables have to be joined with trillions of row together to find a combined table and then run algorithms to find the most relevant information for the user. Hence moving the data from tabular form to (graph) based data structure is brought by NoSQL structures.

NoSQL databases are often very fast, do not require fixed table schemas, avoid join operations by storing denormalized data, and are designed to scale horizontally. The most popular NoSQL systems include MongoDB, CouchDB, HBase, Riak etc which are all open source software products. The data structures used by NoSQL databases (e.g. key-value, wide column, graph, or document) are different from those used by default in relational databases, making some operations faster in NoSQL. The particular suitability of a given NoSQL database depends on the problem it must solve. Sometimes the data structures used by NoSQL databases are also viewed as "more flexible" than relational database tables. NewSQL is a class of modern relational databases that aims to provide the same scalable performance of NoSQL systems for online transaction processing (read-write) workloads while still using SQL and maintaining the ACID guarantees of a traditional database system

5. NewSQL:

NewSQL is a class of modern relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing(OLTP) read-write workloads while still maintaining the ACID guarantees of a traditional database system. Although NewSQL systems vary greatly in their internal architectures, the two distinguishing features common amongst them is that they all support the relational data model and use SQL as their primary interface. The applications targeted by these NewSQL systems are characterized as having a large number of transactions that are short-lived (i.e., no user stalls), touch a small subset of data using index lookups (i.e., no full table scans or large distributed joins) and are repetitive (i.e. executing the same queries with different inputs). These NewSQL systems achieve high performance and scalability by eschewing much of the legacy architecture of the original IBM System R design, such as heavyweight recovery or concurrency control algorithms. One of the first known NewSQL systems is the H-Store parallel database system.

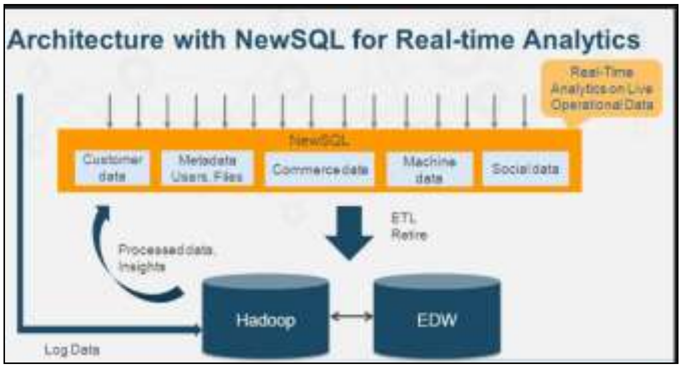


Figure 7: Architecture for NewSQL

ADVANTAGES OF DATABASE MANAGEMENT SYSTEM

DBMS has many advantages when it comes to store and manage data but is also an overhead. One of the biggest advantages of using a DBMS is that it lets end users and application programmers access and use the same data while managing data integrity. It provides a central repository of data that can be accessed by multiple users in a controlled manner. Data is better protected and maintained when it can be shared using a DBMS instead of creating new iterations of the same data stored in new files for every new application. Central storage and management of data within the DBMS provides: data security, Data abstraction and independence, a locking mechanism for concurrent access, an efficient handler to balance the needs of multiple applications using the same data, ability to swiftly recover from crashes and errors, including restartability and recoverability, logging and auditing of activity, robust data integrity capabilities, simple access using a standard application programming interface (API), uniform administration procedures for data and a large amount for forensics investigation of database.

A DBMS can also provide many views of a single database schema. A view defines what data the user sees and how that user sees the data. The DBMS provides a level of abstraction between the conceptual schema that defines the logical structure of the database and the physical schema that describes the files, indexes and other physical mechanisms used by the database. When a DBMS is used, systems can be modified much more easily when business requirements change. New categories of data can be added to the database without disrupting the existing system and applications can be insulated from how data is structured and [5]

ROLE OF DBMS ON GROWING INDUSTRIES

DBMS increases the efficiency of business operations and reduces overall cost. Database management systems are important to businesses and organizations because they provide a highly efficient method for handling multiple types of data. Some of the data that are easily managed with this type of system include: employee records, student information, payroll, accounting, project management, inventory and library books. These systems are built to be extremely versatile. Without database management, tasks have to be done manually and take more time. Data can be categorized and structured to suit the needs of the company or organization. Data is entered into the system and accessed on a routine basis by assigned users. Each user may have an assigned password to gain access to their part of the system. Multiple users can use the system at the same time in different ways.

For example, a company's human resources department uses the database to manage employee records, distribute legal information to employees and create updated hiring reports. A manufacturer might use this type of system to keep track of production, inventory and distribution. In both scenarios, the database management system operates to create a smoother and more organized working environment.

A simple database has a single table with rows for the data and columns that define the data elements. For an address book, the table columns define data elements such as name, address, city, state and phone number, while a table row, or record, contains data for each person in the book. The query language provides a way to find specific types of data in each record and return results that match the criteria. These results display in a form that uses the defined data elements but only shows records that meet the criteria. These three components make up almost every type of database.

Relational databases use multiple tables and define relationships between them using a schema in addition to data elements. Records and data elements from each table merge, based on the query, and display in the form. Routinely used queries often become reports. A report uses the same query but reports on changes in data over time.

SECURITY OF DBMS

Because of the critical importance of database technology to the smooth running of an enterprise, database systems include complex mechanisms to deliver the required performance, security, and availability, and allow database administrators to control the use of these features. Database security deals with all various aspects of protecting the database content, its owners, and its users. It ranges from protection from intentional unauthorized database uses to unintentional database accesses by unauthorized entities (e.g., a person or a computer program).

Database access control deals with controlling who (a person or a certain computer program) is allowed to access what information in the database. The information may comprise specific database objects e.g., record types, specific records, data structures, certain computations over certain objects e.g., query types, or specific queries, or utilizing specific access paths to the former e.g., using specific indexes or other data structures to access information. Database access controls are set by special authorized by the database owner that uses dedicated protected security DBMS interfaces. This may be managed directly on an individual basis, or by the assignment of individuals and privileges to groups, or (in the most elaborate models) through the assignment of individuals and groups to roles which are then granted entitlements. Data security prevents unauthorized users from viewing or updating the database.

APPLICATIONS

A Database management system is a computerized record-keeping system. It is a repository or a container for collection of computerized data files. The overall purpose of DBMS is to allow the users to define, store, retrieve and update the information contained in the database on demand. Information can be anything that is of significance to an individual or organization. Some of the major areas of application are as follows:

(i)Data that is well organized and integrated is very useful in decision making.
(ii)Effective and efficient management of data
(iii) Query processing and management
(iv) Easy to understand and user friendly

(v) Security and integrity of data
(vi) Better Decision making
(vii) Data sharing and storage
(viii) Better access to accurate data
(ix) Ensures error free information

CONCLUSION & FUTURE SCOPE

A database management system is important because it manages data efficiently and allows users to perform multiple tasks with ease. A database management system stores, organizes and manages a large amount of information within a single software application. Use of this system increases efficiency of business operations and reduces overall costs.

Databases form the foundation of analytics industry.

REFERENCES

1. https://en.wikipedia.org/wiki/DatabaseJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.
2. https://en.wikipedia.org/wiki/NewSQLK. Elissa, “Title of paper if known,” unpublished.
3. http://marketrealist.com/2014/04/rising-importance-database management-will-benefit-oracle/
4. http://se-pubs.dbs.uni leipzig.de/files/Andany1991ManagementOfSchemaEvolution.pdf
5. "Evolution of databse systems", by Krzysztof Dembczy´nski