

Evolving databases And it's future aspects

Ajay Kumar Sahu(19111003)

January 2022

1 Abstract

A good database is the first step in establishing a meaningful data. It should be stored in a way that it could become information for the reader and for whoever accessing it. The growth of a database can be determined by the various challenges it faces. Doing so helps researchers develop new ideas and combinations. A data must survive for as long as it is valuable, which means it must be able to adapt in response to the changing demands of the application systems that utilise it. This might include changes in data, meta-data, programmes, and applications, as well as how users perceive the information models. Programs/application are defined by their requirement to store enormous volumes of data, the structure of which must develop as the applications that utilise it. This necessitates the data to be dynamically mapped to a growing schema.

In this paper we will be discussing various phase of DBMS in which it has been evolve to the present status of the database management system and it's scope on various domains.

2 Introduction

A database is a collection of data, typically describing the activities of one or more related organizations. Database management systems may be thought of as a subset of computer science in general. Languages, object-oriented programming, and other programming paradigms, compilation, operating systems, concurrent programming, data structures, algorithms, theory, parallel and distributed systems, user interfaces, expert systems and artificial intelligence,

statistical techniques, and dynamic programming are among them.

A database management system, or DBMS, is software that makes it easier to manage and use huge amounts of data. A database is a collection of information that usually describes the activity of one or more organisations. The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized. The demand for such systems, as well as their application, is continuously increasing. Instead of employing a database management system, you may store the data in files and manage it with application-specific code. The user of a database management system (DBMS) is ultimately concerned with a real-world business, and the data to be stored defines different parts of that business. It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

When any of a table's characteristics has redundancy in values, DBMS improves data organisation by using a database schema design approach called normalisation, which separates a large table into smaller tables. DBMSs provide a number of advantages over traditional file systems, including greater flexibility and a more sophisticated backup mechanism.

3 Different phases of evolution in DBMS

Charles Bachman invented IDS (Integrated data store), a network database, in the early 1960s, and it impacted other DBMS for future progress. It was created to boost performance using the hardware available at the time. IBM launched the IMS (Information Management System) in 1966, which was based on a hierarchical database. It was created to store stock information for very large bills of materials for the Saturn V moon rocket and the Apollo spacecraft.

E. F. Codd introduced the relational data model in 1970. For the model, it was a theoretical paper that was presented. Following that, E. F. Codd delivered a series of papers. According to him, data in a relational database is represented in table form. The data is accessed without the use of an algorithm, and a non-procedural

high-level language is employed. The Relational Model was introduced as a standard technique for DBMS in 1980. SQL is a query language that has been accepted by ANSI and ISO. Many relational DBMS, such as Oracle, DB2, and Informix, were introduced in the same year. Researchers discovered that relational models do not accommodate multimedia data, unstructured data, or inheritance relationships after adopting RDBMS. The relational paradigm does not facilitate application scaling. When numerous tables are linked together, query processing becomes inefficient. The cost of linking tables is considerable because RDBMS join operations are costly.

The object-relational database was created in 1990. The object-relational database concept combines the object notion with the relational database concept. It's like combining relational databases with OOP principles (inheritance, encapsulation, and polymorphism). This concept was introduced by Stonebaker et al., who advocated that RDBMS capabilities be expanded by supporting rich object structures. Data structures (object table for storage), Integrity constraints (object identification, relationship), and Operations are all essential components for an object relational model. Object-Relational's major purpose is to bridge the gap between relational databases and object-oriented modelling techniques used in programming languages like Java and C++. While developing applications, in order to achieve storage for complicated data types and to display relationships. A user may obtain transaction management, scalability, flexibility, performance management, user-defined data types, and much more by combining these two technologies. Increased complexity, as well as the expense that comes with it, there is a significant semantic gap between two technologies. Object applications, unlike relational-based systems, are not data-centric.

The OODBMS (Object- Oriented Database management system) was created around 1990. The idea behind OODBMS is to combine the principles of object-oriented programming with database administration. Encapsulation, complex objects, types and classes, extensibility, persistence, and other characteristics are supported. In an OODBMS, data is represented in the form of classes and objects rather than records, as in the E-R paradigm. Objects are run-time entities that play a critical role in real-world problems. All values

in an object are kept in the form of instance variables. The term "class" refers to a group of similar objects with similar values and methods. By invoking the methods of other objects, two objects can interact with each other. Because the method and data are not visible outside the object, it provides data abstraction. It was designed to store and manage object created by programming languages like java, C++ and provide object-oriented features to users. It was designed to satisfy the demand for better model for real world entities and to provide richer data model than traditional data model.

There was a lack of flexibility with an OODBMS, as each change in the schema forces the user to alter other classes as well. It was language-dependent since the API was bound to a certain language. The queries that can be run on the data in an OODBMS are heavily reliant on the system's architecture. Other difficulties included a lack of security, a rise in complexity, a lack of standards, and a lack of experience. OODB does not support Views, despite several proposals. It was difficult to create an object-oriented view using a model that had an object identification feature.