A GENERAL APPROACH OF DATABASE SCHEME AND ITS COMPARATIVE STUDY

Sanjeev Kumar Mandal¹, Bhavana Gowda D M², Lalit Kumar³, Menaka C⁴, T Reshma⁵
Deepika N⁶

¹,²,³Assistant Professor, Dept of BCA, School of Computer Science & IT, Jain Deemed to be University, Bengaluru, India
⁴Associate Professor, Dept of BCA, School of Computer Science & IT, Jain Deemed to be University, Bengaluru, India
⁵,⁶Research Scholar, Dept of BCA, School of Computer Science & IT, Jain Deemed to be University, Bengaluru, India

ABSTRACT

The usage of database originates back to the early 60s. Since then its growth is determined in different aspects. Different demands of every era give database a new bunch of challenges. To overcome these challenges, researchers come up with different ideas and methods. These various combinations enhance features of database and this way database starts evolving from one period to another. Due to this reason the database that we had in 1960 is completely different from what we have now. This review paper briefly explains the types of database and the database services provided by Firebase.

Keywords: Database, Types of DB, Relational DB, Object oriented DB, Graph DB, Distributed DB, NoSQL DB, Cloud DB, Firebase, Cloud Firestore, Realtime Database.

I. INTRODUCTION

A database is a collection of organized information, which can be easily accessed, managed and updated. Computer databases typically contain aggregations of data records or files, containing information about sales transactions or interactions with specific customers [1].

1 Types of Databases

Database has metamorphosed since its commencement in the 1960s, dawning with hierarchical and network databases, through the 1980s with object-oriented databases, and present day with SQL and NoSQL databases and cloud databases [2]. Databases can be classified according to their content type: bibliographic, full text, numeric and images. In computing, databases are sometimes classified according to their organizational approach. There are many different kinds of databases, ranging from the most prevalent approach, the relational database, to a distributed database, cloud database, graph database or NoSQL database [1], [3].
1.1. Relational Database

These databases are organised as a set of tables, in which data is stored into a pre-defined category. Each table consists of rows and columns where the column has an entry for data for a specific category and rows contains instance for that data defined according to the category. The intersection of row and column is called as a cell. The Structured Query Language (SQL) is the standard user and application program interface for a relational database [4].

There are various simple operations that can be applied over the table which makes these databases easier to extend, join two databases with a common relation and modify all existing applications.

1.2. Object Oriented Database

An object-oriented database is an amalgamation of object-oriented programming and relational database. There are various elements which can be created using object-oriented programming languages (like C++, Java) and that can be stored in relational databases, and object-oriented databases are well-suited for those items. An object-oriented database is organized around objects rather than actions, and data rather than logic. For example, a multimedia record in a relational database can be a definable data object, as opposed to an alphanumeric value [3], [5].

1.3. Distributed Database

The distributed database has endowments from the common database as well as the information captured by local computers also. The data unlike the customary, is not stored at one place and is distributed at various sites of an organization. These sites are connected to each other with the help of communication links which helps them to access the distributed data with ease.

There are two kinds of distributed database: homogenous and heterogeneous [4]. The databases which have same fundamental hardware and run over same OS and application procedures are known as homogeneous DDB. Whereas, if the OS, underlying fundamental hardware as well as application procedures is different at various locations of a DDB, then it is known as heterogeneous DDB [2], [6].

1.4. Cloud Database

A cloud database is a database that has been optimized or built for a virtualized environment. They can be categorized either in a hybrid cloud, public cloud or private cloud. Cloud databases provide benefits such as the ability to pay for storage capacity and bandwidth on a per-use basis, and they provide scalability on demand, along with high availability. A cloud database also gives enterprises the opportunity to support business applications in a software-as-a-service deployment [7], [8].

1.5. Graph Database
The graph is an assemblage of nodes and edges, where each node represents an entity and each edge describes the relationship between entities. A graph-oriented database, or graph database, is a type of NoSQL database that utilises graph theory to store, map and query relationships.

Graph databases are primarily used for analysing interconnections [9]. For example, companies might use a graph database to mine data about customers from social media.

1.6. NoSQL Database

These are implemented for large sets of distributed data. These are very efficient in analyzing large size unstructured data that may be stored at multiple virtual servers of the cloud. Cloud Firestore and Realtime Database provided by Firebase platform of Google are few examples of no-SQL database [1].

2. Firebase Platform

Firebase is Google’s mobile application development platform that helps you build, improve, and grow your app. It provides features like analytics, authentication, databases, configuration, file storage, push messaging, and many more. The services are hosted in the cloud, and can be scaled with little to no effort on the part of the developer. The products have backend components that are fully maintained and operated by Google. Client SDKs which are provided by Firebase interact with these backend services directly, with no need to establish any middleware between the app and the service.

This is different than traditional app development, which usually involves writing both frontend and backend software. The frontend code just invokes API endpoints exposed by the backend, and the backend code actually does all the work. However, with Firebase products, the traditional backend is bypassed, putting the work into the client. Administrative access to each of these products is provided by the Firebase console Databases provided by Firebase include: Cloud Firestore and Realtime Database [1].

2.1. Cloud Firestore

Cloud Firestore [7] is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps - at global scale. Cloud Firestore ships with mobile and web SDKs and a comprehensive set of security rules so you can access your database without needing to stand up your own server. Using Cloud Functions, we can execute hosted backend code that responds to data changes in database.

We can access cloud Firestore with traditional client libraries too (i.e. Node, Python, Go, and Java). With Cloud Firestore, we can automatically synchronize your app data between devices. The users can access and make changes to their data at any time, even when they're offline. Offline mode is available on iOS, Android and Web!
We can restrict data access based on user identity data, pattern matching on your data, and more. Cloud Firestore also integrates with Firebase Authentication to give a simple and intuitive user authentication.

2.2. Advantages of Cloud Firestore DB

- Combines the Benefits of Firebase with Google Cloud Platform
- Cloud or Server-less Solution
- Offers Excellent Data Handling Capabilities
- Designed to Scale
- Promises Robust Security
- Enables Offline Support
- Cost-Effective Pricing

2.3. Realtime Database

The Firebase Realtime Database[^9] is a cloud-hosted NoSQL database that lets you store and sync data between users in realtime. It allows realtime syncing which makes it easy for users to access their data from any device: web or mobile, and it helps users collaborate with one another. The Realtime Database integrates with Firebase Authentication to provide simple and intuitive authentication for developers. When the users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. Realtime Database can be used with mobile and web SDKs, which allows developers to build apps without the need of servers.
2.3.1 Advantages of Realtime DB

- Offline Synchronization
- Real-time Time Updates
- Data splitting and scaling
- Access from different devices
- Pay as grow go pricing

3. Comparative study of different types of Databases

<table>
<thead>
<tr>
<th>Feature</th>
<th>Data Structure</th>
<th>Version</th>
<th>Licensing</th>
<th>Documentation</th>
<th>Developer &amp; Initial release</th>
<th>Security Management</th>
<th>Multi User Access Control</th>
<th>Backup Mechanism</th>
<th>Transaction (Support to ensure data integrity after non-atomic manipulations of data)</th>
<th>Implementation language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation Database</td>
<td>BTree and hash index structure</td>
<td>8.0.23 (18 January 2021)</td>
<td>GNU General Public License</td>
<td>✓ <a href="https://dev.mysql.com/doc/">https://dev.mysql.com/doc/</a></td>
<td>Swedish company MYSQLLAB, 23 May 1995</td>
<td>MySQL Enterprise TDE uses industry standard AES algorithms.</td>
<td>Users with fine-grained authorization concept</td>
<td>It offers only two backup mechanisms that are mysqlhotcopy and mysqldump.</td>
<td>ACID</td>
<td>C and C++</td>
</tr>
<tr>
<td>Object Oriented Database</td>
<td>Object-relational mapping</td>
<td>6.7.1 (October 2018)</td>
<td>Proprietary Commercial</td>
<td>✓ <a href="https://downloads.gemtalks.com">https://downloads.gemtalks.com</a></td>
<td>Servio Logic, March 1,</td>
<td>TGS secret key for hashing</td>
<td>GemStone one can support over</td>
<td>It supports 3 types of</td>
<td>ACID</td>
<td>Smalltalk</td>
</tr>
</tbody>
</table>

www.turkjphysiotherrehabil.org
<table>
<thead>
<tr>
<th>Distributed Database Ex: Oracle</th>
<th>Logical storage structures</th>
<th>21c F36089-13 (March 2021)</th>
<th>Proprietary commercial software</th>
<th>Oracle corporation, 1979</th>
<th>DBMS CRYPT O PL/SQL package supports asymmetric key operations: PKENC RYPT PKDEC RYPT SIGN VERIFY</th>
<th>Fine grained access rights according to SQL-standard</th>
<th>It offers many backup mechanisms that are backup, hot backup, import, export, etc.</th>
<th>ACID C and C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Database Ex: Nebula Graph</td>
<td>Native graph database and multi-model graph database</td>
<td>2.0.0-alpha (November 2020)</td>
<td>Apache 2.0, Open Source, Common Clause 1.0</td>
<td>Yesoft Inc., 2019</td>
<td>Lightweight Directory Access Protocol (LDAP) authentication</td>
<td>Role-based access control</td>
<td>Nebula Graph provides a scheduled cold backup mechanism</td>
<td>ACID C++</td>
</tr>
<tr>
<td>No SQL Database Ex: Cloud Firestore</td>
<td>NoSQL, document-oriented database</td>
<td>CLI v.9.7.0 (March 24, 2021)</td>
<td>Proprietary commercial software</td>
<td>GoogLe Cloud, 2017</td>
<td>256-bit Advanced Encryption Standard, and each encryption key is itself encrypted</td>
<td>Access rights for users, groups and roles based on Google Cloud Identity</td>
<td>Google Cloud</td>
<td>Yes -</td>
</tr>
</tbody>
</table>
d with a regularly rotated set of master keys.

<table>
<thead>
<tr>
<th>Cloud Databases</th>
<th>Cloud based relational databases</th>
<th>Proprietary commercial software</th>
<th>Microsoft, 2010</th>
<th>It lets software developers encrypt data by using AES and 3DES encryption algorithms without changing existing applications</th>
<th>Fine grained access rights according to SQL-standard</th>
<th>No, only system-initiated automatic backups - see Automated backups</th>
<th>ACID</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Azure</td>
<td></td>
<td><a href="https://docs.microsoft.com/en-us/azure/azure-sql/">https://docs.microsoft.com/en-us/azure/azure-sql/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## II. CONCLUSION AND FUTURE WORK

In this paper, a general approach of database practices is presented. Then, a comparative study is made between the different types of databases. Furthermore, analysis and comparative study is done in order to provide a satisfying level in terms of data storage and time. Since the research on database storage is a relatively young area, the number of new formulate database is increasing as long as many attacks are appearing so direction of the future research work is about verifying their efficiency.

## REFERENCES