**PROG8410-23S-Section 2 – NoSQL Database Implementation**

**LAB 1 Assignment**

1. When I was researching an interesting fact about DBMS online. One fact pushed me to know more about it. It looks like these database concepts were initially used by military agencies and this was news to me and made me understand how critical these database concepts were. The term “Database Management system” was introduced in the late 1960s and it was also referred to as a “Database”. These concepts were popularized beyond the world of military agencies only after the two symposia held in 1964 & 1965. These events were organized by The System Development Corporation (SDC) of California in collaboration with military agencies. In the 1970s – these were promoted and sold to corporate entities as a technology that helped a company integrate all its data to a digital data in a single place. The usage of the term “Data Base Management System” started from 1971 onwards soon after it was adopted by the DBTG, The Data Base Task Group (DBTG), chaired by William Olle of RCA. It was an ad-hoc committee of the computer industry group CODASYL (Committee on Data Systems Languages). Now, these “Database Management Systems” have developed gradually from a class of programs known as “File management systems”.

Citation - <https://aequortechnologies.wordpress.com/2015/08/21/10-interesting-facts-about-database-management-systems/>

1. ‘Metadata’ term is defined as data about data which helps to organize, find, and understand the provided data. Basically, there are main major categories of Metadata. When it is referred to, in DBMS, it is consistent with their uses and domain.
   1. Those types and its corresponding examples are as follows:

* **Technical Metadata** – This metadata defines the information about the database system. For E.g.: This metadata includes database system names, table names, table size, data types, records, and attributes. Furthermore, it includes the applicable constraints like indexes, primary key, and foreign key.
* **Business Metadata** – This metadata defines the information about the specific business. For E.g.: This metadata includes the ownership of data, changing policies, business rules and regulations and other business details.
* **Descriptive Metadata** – This metadata describes the information about any file, folder, book, image, or video. This information includes how, when, why, who, where and how big the data is. For E.g.: If a book has been considered then it includes the details of knowledge about the title, author, date, size, author name, publisher name, publication details, and price.
* **Operational Metadata** – This metadata includes the information, which is currently under any operation and furthermore, it acts as the data which is used by execute-level managers to track, manage, and perform any task. For E.g.: If a data warehouse has been considered then this metadata would be the roadmap that defines its object, and it becomes the directory for the decision support system to locate the content of the data warehouse based on its need.

Citation –

<https://www.geeksforgeeks.org/metadata-in-dbms-and-its-types/>

1. I have considered the Atomicity property from ACID for my real-life example in which the customer pays their individual health insurance premium amount online from their bank account to their corresponding insurance company’s account against their active policy which falls under the insurance domain.
   1. The reason for choosing Atomicity property is that there should be only a single transaction in which the amount has to be debited from the customer’s account and credited to the insurance company’s account. Mainly, either the transaction should happen and update the bank accounts appropriately else it must roll back the whole transaction and come back to the previous state in both accounts.
   2. There are a few unavoidable drawbacks to this Atomicity property:
      1. Since multiple transactions are grouped under a single transaction. Until the whole transaction commits or aborts, it acquires the lock state. Thus, giving an opportunity for a deadlock within a single transaction.
      2. In every rollback, the performance and efficiency will be impacted.
      3. This application is using a distributed database that includes multiple servers therefore, it is difficult to implement.
      4. Maintenance cost is high and harder to upgrade.
   3. At the same time, there are multiple benefits when using this Atomicity property:
      1. The transactions are performed in a timely and efficient manner.
      2. The transaction of financial data is safe and secured by maintaining the integrity of the data.
      3. The data is consistent in case of any failure since the previous state has been withdrawn.
      4. The implantation can be done with in the stipulated time.
      5. It establishes the reliability of data.
      6. If the transaction is complex, then it can be split into many smaller transactions and can be executed as a single transaction.

Citation –

Self, <https://docs.oracle.com/cd/E17275_01/html/programmer_reference/transapp_atomicity.html#:~:text=Atomicity%20means%20that%20multiple%20operations,or%20none%20of%20the%20changes>.

<https://www.beekeeperstudio.io/blog/what-is-atomicity-in-databases#:~:text=Finally%2C%20atomicity%20helps%20to%20ensure,part%20of%20the%20transaction%20fails>.

1. The **CAP theorem** also known as Brewer’s theorem states that a distributed database system can only guarantee two out of these three characteristics: Consistency, Availability, and Partition Tolerance.
   * Consistency – All nodes should have the same data and when retrieved all nodes should send the most recent updated data.
   * Availability – The system remains up and running all the time.
   * Partition Tolerance – The system should not fail even if there is any issue in one node.
2. The benefits of CAP theorem would be:
   1. It allows the architect to analyse and come up with the effective system design when selecting the distributed database based on the specific business requirements.
3. The drawback of CAP theorem outweighs its benefits because of the ambiguity in its definition. Those shortcomings are as follows:
   1. There is a single narrow trade-off between the consistency and availability. There is no way to balance between consistency and availability.
   2. In the definition there is no quantitative measure for network latency.
   3. In case of network partition, even if one node fails then the availability is impacted.

Citation –

<https://www.educative.io/answers/what-is-the-cap-theorem?utm_campaign=brand_educative&utm_source=google&utm_medium=ppc&utm_content=performance_max&eid=5082902844932096&utm_term=&utm_campaign=%5BNew%5D+Performance+Max&utm_source=adwords&utm_medium=ppc&hsa_acc=5451446008&hsa_cam=18511913007&hsa_grp=&hsa_ad=&hsa_src=x&hsa_tgt=&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=EAIaIQobChMI6LWD6IKI_wIVXt_jBx3M-AqzEAAYASAAEgLXKvD_BwE>

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.ijcaonline.org/archives/volume151/number10/patinge-2016-ijca-911921.pdf

By Vidhya Venugopal

Thank you!