Assignment - 5
CSCI 5408 - Data Management, Warehousing, Analytics

Summary: The increase of need for better handling of big data, especially for web applications, has led to the popularity of no-SQL or not-so-SQL databases compared to relational databases. RDBMS enforces a highly structured schema and is not horizontally scalable, just by increasing the existing number of nodes. In contrast, NoSQL databases can store structured, semi-structured, and unstructured data and are horizontally scalable depending on the business needs. The flexibility of NoSQL data stores such as MongoDB and Cassandra has made them a go-to database for most applications. However, this research paper further gives a better comparison between SQL and NoSQL databases. Further, it emphasizes the security implication on NoSQL databases, with steps to take for better security and privacy solutions.

The pretext for Security and Privacy solutions: Without a doubt, the use of NoSQL databases allowed the storage of unstructured data improved the system's scalability, and reduced the cost of storage. However, this does not mean the use of RDBMS is going away any time soon. The privacy of the data and for performing queries on encrypted data was of significant concern in NoSQL. To begin with, the earlier research for better security and privacy solutions for NoSQL databases talks about using a proxy service in a trusted environment to rewrite the queries and perform computation over encrypted data in untrusted resources. One such example was performing search operations on privacy- preserving key-value stores on top of the Redis database.

Comparison between SQL and NoSQL:

Topic	**Relational Database (SQL)**	**Not so SQL (NoSQL)**
Reliability of Transactions	ACID (Atomicity, Consistency, Isolation, and Durability).	Does not provide full ACID support - eventually consistent.
Scalability Issues and Cloud Support	● Horizontally Scalable. ● Fully compatible with the cloud environment. ● Supports data search on the full content. ● Supports unstructured, semistructured, and structured data.	● Vertically Scalable and requires additional work for horizontal scaling. ● Compatible with the cloud environment. ● Do not provide data search on full content. ● Supports only structured data.
Complexity and Big Data Management	● Less complexity. ● Abstract model - Creating a schema is not always required. ● High speed of retrieving and storing data in distributed nodes.	● Higher complexity. ● Structured model - Schema is always defined. ● Accuracy is more important than speed.
Data Model	● A graph-like data structure. ● Do not use the table as a storage structure.	● Sets like data structure ● Data is stored in tables and has a subset of the Cartesian product as

	• Schema-less - supports unstructured data like word, pdf, images, and video files.	the relationship. • Columns and rows are predefined in the schema.
Data Warehouse and Crash Recovery	• Not ideal for Big data. Focuses on scalability, availability, and high performance • It depends on replication for crash recovery.	• Not ideal for Big data. Oversize of storage results in Big data problems • Crash recovery is handled by the recovery manager, primarily using logs.
Privacy and Security	• Serious shortcomings of NoSQL is the lack of security. • Low transparency and only a few categories of NoSQL offer data protection.	• Most databases do not offer to inherit security and privacy features. • Commonly used algorithms for securing communication and ensuring data confidentiality in relational databases.

Security and Privacy Solution: In an anonymous system, credentials of multiple users are initially inserted on connection, and various transactions are not tied to a user. Hence, it has the best security for users based on RSA (Rivest–Shamir–Adleman) and the Diffie Hellman protocols. The components are Users (U), Central Identity Provider (IP), Service Providers (SP), and the Organization for credentials. In addition, the IP has its public and secret key to sign sensitive data. A user can prove ownership to the Organization by revealing ownership details. Furthermore, CA (Credential Authority) ensures every user is unique, alongside the Verifier (V), checks the validity and verifies the user. On the other hand, organizations are independent and select their public and secret key to better key management. However, the Organization can only know the ownership as proved by the user and nothing else. On the user-side, SP can reveal and blacklist the user to prevent abuse of the service. It is clear from these interconnections that the users' privacy is at par without having complete control under SP and IP.

The Kerberos central authentication system can be bypassed with the help of advanced scripts, and monitoring is usually limited to data processing. In addition, the lack of log files and information of communication between the nodes in a cluster makes it more challenging to identify a data breach or data loss. While certain tools offer better security features for monitoring big data systems, there is no clear definition for usage; the most common is early authentication using Kerberos, followed by the second level of authentication to access MapReduce.

In conclusion, the research papers discuss the security and privacy concerns in NoSQL databases and further emphasize the use of Kerberos for authorization and nodes and TDE in the case of Cassandra. Finally talks about the various attacks and steps to mitigate them on NoSQL databases.

Reference:

G. Vonitsanos, E. Dritsas, A. Kanavos, P. Mylonas and S. Sioutas, "Security and Privacy Solutions associated
with NoSQL Data Stores," 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA, 2020, pp. 1-5, DOI: 10.1109/SMAP49528.2020.9248442.