# Database Backup and Recovery: A Review with test implementation for MySQL and NoSQL Databases

N. Fathima Nifra[1], MS. Suhail Razeeth[2]

[1,2]*Dept. of Information and Communication Technology, South Eastern University of Sri Lanka*

[1]fathimanifranizar6@gmail.com, [2]razeethsuhail@seu.ac.lk

***Abstract***

*Databases hold cloud data and applications, and cloud-hosted services and apps must support more data. Therefore, the backup and recovery capabilities of the database are crucial. Regular backups of structured and stored data can reduce data loss and corruption. Administrators restore the database from a backup in the event of a hardware failure, data loss due to bugs, or other issues. The correct methods and features are important to the database administrator and user to apply the right technique at right time. The aim of the study incorporates identifying the available techniques of database backup and recovery with the implementation method. As a result, we can distinguish between the types of backup such as full, incremental, and differential, as well as recovery techniques such as log recovery and shadow paging. Additionally, the results indicated the adoption of MySQL and NoSQL backup and recovery methodologies. The result also showed the implementation of backup and recovery techniques of MySQL and NoSQL. This could be a solution to database backup and recovery issues if we use the appropriate approach and functionality.*

***Keywords: Database System, Data Management, Database backup, Recovery, mysqldump, mongodump***

## I. INTRODUCTION

Database systems (Ko et al., 2021; Li et al., 2021; Bao et al., 2022; Hu, 2022) are commonly used to store data in cloud services and applications (Heidari and Navimipour, 2021; Khan et al., 2021; Mishra, Sharma and Alowaidi, 2021; Sahu, Raghavan, and Chandrasekaran, 2021). Data backup and recovery (Bohora et al., 2021; Jing et al., 2021; Zhang, Xu, and Muntean, 2021; Zhang et al., 2022) is an important function of the database system. Periodic backups of structural information (e.g., schema, indexes, and so on) and stored data help database managers and database administrators (Aleryani, no date) avoid data loss and corruption. Administrators restore the database from a backup in the event of hardware failure, data damage due to bugs, system problems, or other circumstances. Cloud-based services and apps must back up expand data.

Many cloud providers execute regular database backups and recovery in order to ensure the safety of their customers' data in the event of a breakdown (Chang et al., 2021; Jiang et al., 2021; Maher and Nasr, 2021). A failure is a situation when the system doesn't operate as expected. Some failures are the result of hardware problems (such as a power outage or disk failure), software problems (such as program flaws or incorrect data), or human mistakes (For instance, an operator mounting the incorrect tape on a drive or an inadvertent user action.) Failure happens when the system's algorithms attempt to process an incorrect state. Physical and logical backup and recovery are common in database systems. The physical technique uses raw data by copying database files, while the logical technique extracts data as a sequence of query statements. Both solutions offer full backup and restore of the entire database and incremental backup of updated data since the last backup. Both backup options are slow and need extra I/O operations.

A physical technique, which duplicates files containing the database's architecture and table files, is one of the ways. The physical technique is intuitive since it involves duplicating the database's files for backup and recovery. A logical technique, on the other hand, analyzes the whole database for sequences of query statements that may be used to recreate the database. Backup and Recovery data is portable between file systems (Cameselle and Labrador, 2021), operating systems (Banecs, Babarada, and Ravariu, 2020), and MySQL versions since the logical technique are based on query statements.


*Faculty of Technology, South Eastern University of Sri Lanka (ISBN 978-624-5736-40-9)*

### A. IMPORTANCE OF BACKUP AND RECOVERY

The backup copy can be retrieved if the main data is lost. An attack (virus or malware), accidental data erasure, or data corruption can cause primary data failures. Hardware or software faults can generate these events. Backup copies allow data recovery from an earlier time, which helps a company recover after an unanticipated event.

In order to protect against the loss or distortion of the original data, it is required to keep a copy on a secondary medium. As simple as a USB stick or external hard drive, as complex as a disk storage system, cloud storage container, or tape drive, this additional medium can be used to store data. There is no right or wrong when it comes to where you store your backup media. Keep copies of data from weather-related incidents in remote locations.

Regular backups reduce data loss between backups. The time between backups increases the possibility of data loss when recovering. Multiple data copies allow you to restore to a time before data damage or malicious attacks.

### B. DATABASE BACKUP AND RECOVERY TECHNIQUES.

There are p and logical strategies for backup and recovery databases. Physical backup uses raw data, while logical uses query statements. As proposed, both techniques are at the application layer. This section explains backup and recovery mechanisms.

#### 1) Database Backup

A database backup is a method of periodically establishing the second instance of a database. A DBA can perform this task manually, with the use of a backup script, an automated service, database administration software, or by hand. The database can then be retrieved in full or in part from the backup. Backups can be classified as either logical or physical. Physical backups are made up of copies of the original database files. When it comes to logical backups, the Oracle Export tool is used to extract and store binary data. In addition to physical backups, logical backups can be used. There are two main reasons to keep backups. Restoring a system's state is the first step in disaster recovery, whereas restoring specific files that were accidentally deleted or corrupted is the second. (Kadry et al., 2011).

There are three main methods for backing up databases: full, incremental, and differential. Organizations should employ a solid mix of at least two of these techniques to reduce downtime.

##### i. Full Database Backup

A full backup is the most fundamental and comprehensive kind of backup process. When a full database backup is conducted, all the database's data is saved. For example, Xtrabackup of Percona [6] makes use of a physical backup to give a backup solution. Xtrabackup creates a complete backup of your database by copying all its files (Kim, Yeom, and Son, 2020). Complete backups can copy all data, including user data, system files, and configuration information. This backup replicates data to a disk or tape. Duplicating everything is slow. Complete backups for every operation store all data on a single set of media. This statistic is called a recovery time target. It takes longer to conduct a full backup and requires more storage space.

This backup method is typically done periodically for large data centers, however, if the database is a small one, this type of backup could be done on a daily or even a more frequent interval. This strategy guarantees that all data is in one location from a given date even when a full backup takes some time. The procedure can thus be continued with a substantial amount of data even if there is some data loss during the restoration process.

Full backups are run periodically. Small data centers (especially those with critical applications) may do a full backup daily or more often in some cases. Complete backups are often used alongside incremental or differential backups.

##### ii. Incremental backup

Incremental backup copies or scans new or altered data. The incremental backup compares each database page to the most recent backup. Xtrabackup puts altered pages in delta files (Kim, Yeom, and Son, 2020). Incremental backups iteratively replicate updated or new files since the last full backup. This backup compares state changes since the last incremental backup. This backup is suitable for restoring recent changes in manageable portions. An organization compares the altered time stamp on files to the most recent backup's time stamp. Backup software records the date and time of backups to track file changes.

Incremental backups take less time and data. An organization can conduct incremental backups as often as needed and only store the most recent changes since they only duplicate changed data since the last backup. Incremental backups duplicate less data than full backups. This will speed up backups and save space. Full and incremental backups are needed to restore a database if all data is lost. In this scenario, we set up a weekly incremental backup schedule with level 0 on Sunday, level 1 on Wednesday, and level 2 on other days (Li and Xu, 2009).

*iii. Differential backup*

This backup method keeps the most recent updates since the last full backup. Differential backups include all data that has been modified since the last full backup. Differential backups improve backup and restore times. Differential backups are faster and smaller than complete database backups because they record just updated or new extents (Kadry et al., 2011). Differential backup stores accumulated changes between initial and daily transaction backups. After a differential backup reaches a specific size, a full backup should be run; otherwise, backup copies will be larger than the baseline. Because of differential backups, only a full backup is needed for restoration. This strategy requires more room and network capacity to check the most recent data.

First-time differential backups act like incremental backups, copying all changed data since the last backup. After that, it copies all data changed since the last full backup. It saves less data than a full backup but more than an incremental backup over time. Differential backups take longer and consume more space than incremental backups, but less than complete backups.

*2) Database Recovery*

Database recovery restores the destroyed, corrupted, or inaccessible data. Data recovery in enterprise IT means restoring data from a backup to a computer, server, or external storage system (Sharma et al., 2012). It is the process of recovering or restoring database data after a system crash, hacking, transaction errors, coincidental damage, infection, unexpected failure, wrong command implementation, etc.

Data saved in databases should always be available, despite data loss or errors. The database must provide tools for quick data recovery. Atomicity means either the transactions exhibit the result of successful completion permanently in the database, or the transaction must have no trace of achievement in the database. Data backup and recovery might be voluntary or automatic. Postponed updates, fast upgrades, and data backups can prevent database loss.

Log-based Recovery and shadow paging can recover databases. Reusing the latest pages speeds up recovery for applications using locality. Though the system fails again during recovery, the database stays consistent because Backup List pages are still in stable memory (Choi et al., 2000). Table 1 below also shows the research Objectives and relevant motivations that were set up for the purpose of this review article.

Table 1: Research Questions and Motivation

| S.No | Research Question | Motivation |
|---|---|---|
| RQ1 | Available Feature of Database backup and Recovery | Recognize the backup and recovery functions as a benchmark for designing a database backup and recovery system that supports business needs. |
| RQ2 | Supported technologies | Identifying the best tools to cut down on the time and money needed for backups and recoveries |
| RQ3 | Security issues in Database backup and recovery System | Examine the causes of any backup/restore security problems and provide improved approaches if possible. |
| RQ4 | Backup and Recovery methods of different database | To assist database users on how to effectively backup and restore their data. |

Here's how the paper continues. A review identifies and describes database backup and recovery solutions. After discussing the review's findings to propose research views, we conclude our study.

## II. LITERATURE REVIEW

Karina Bohora, et al. (Bohora et al., 2021) reviewed Cassandra's backup and recovery procedures. The study analyzed state-of-the-art literature on the topic and discussed Cassandra's deletion method, backup and recovery difficulties, and the study also discusses node and data center failure detection and handling. Also reviewed are backup and recovery options, including disaster recovery.

The research of Hwajung Kim et al. (Kim, Yeom and Son, 2020) is important for database backup and recovery. The study focused on write-ahead logging (WAL) for database backup and recovery. They created a backup solution that uses existing log data without new I/O operations. Using many threads, we optimized WAL's crash recovery mechanism. MySQL is the implemented and evaluated scheme. Experiments show that the suggested technique provides instant backup by eliminating I/O processes. The suggested scheme recovers faster than existing schemes. Yongseok Son et al., (Son et al., 2017) did another comprehensive survey examined the SSD-assisted backup/recovery (Jung and others, 2019; Baek et al., 2020; Dagnaw, Hua, and Zhou, 2020) Database backup and recovery technique using flash-based SSDs (BRSSD). The backup/recovery feature employs a Samsung SM843Tn enterprise-class SATA-based SSD. They later used BR-SSDs on PostgreSQL and MongoDB (Sharma, Sharma, and Bundele, 2018; Makris et al., 2019, 2021; Zimányi, Sakr and Lesuisse, 2020; Woltmann et al., 2021), and different workloads, backup scenarios, and environments were used.

Data backup and recovery are essential to any recovery plan. Backup and recovery are strategies and methods for preserving the database from data loss and regenerating it after data loss. Infrastructure and technical solutions influence database recovery. Some recommendations bridge business-oriented studies with disaster recovery facility design (Cegiela, 2006). In the event of a database recovery, TAIL LOG is used. This option backs up the transaction log and enters recovery mode. Restore all backups. When migrations demand a minimal level of lack, this method is utilized (Gotseva, Gancheva and Georgiev, 2011). Many academics have studied offering backup and recovery across storage levels. For example, ext3cow (Peterson and Burns, 2005) and BTRFS (Rodeh, Bacik and Mason, 2013) backup and restore data using snapshots and copy-on-write (CoW) methods. In addition, numerous researchers have investigated changing the flash translation layer (FTL) of flash-based solid-state drives (SSDs) to provide backup and recovery capabilities on the storage device itself (SSDs) (Huang et al., 2012). Snapshot-based file system backup interferes with normal operations, since ordinary SSDs lack backup and recovery functionalities. To increase recovery and run-time performance, SSDs can be used instead of hard disks (HDD) (Son et al., 2017). Flash-based SSDs provide fast throughput and low latency, making them suitable for data-intensive apps. Using WAL log data to backup databases without adding extra queries or I/O activity shortens recovery times. The suggested scheme on MySQL 8.0.15 analyzes backup and recovery using sysbench. Hwajung Kim et al. (Kim, Yeom and Son, 2020) Storage device performance was assessed for database backup and recovery. This work is crucial for database backup and recovery procedures.

## III. METHODOLOGY

### *A. Systematic Literature Review*

In order to accomplish our targets, we used a systematic methodology to conduct our research. Various learning strategies have been thoroughly investigated in order to safeguard the backup and recovery system. This study was based on research articles in the field of database management system backup and recovery techniques. Figure 04 below shows the flowchart of paper selection.

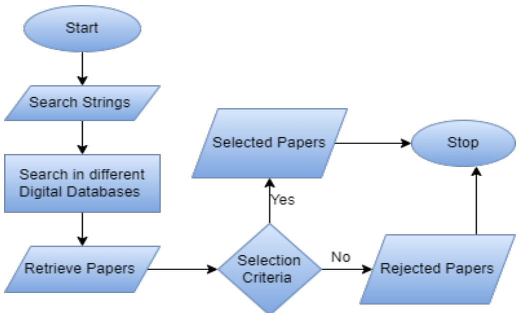


Figure 1: Flowchart for paper selection

This research examines the available literature to learn about the latest innovations and backup and recovery techniques. This technique outlines how to find, understand, and analyze research articles, making it easier to find supporting data. The search plan included expert planning and string validation. From the search results, peer-reviewed and high-quality database publications and conferences were filtered to explore database backup and recovery techniques.

The search phrases were carefully crafted in response to the research question. The search keywords were adjusted several times in order to assemble practically all the relevant papers. As a result, several search strings with different combinations of words were utilized to find relevant papers. "Database backup" AND

"Recovery Techniques" AND "Physical backup and logical backup" OR "MySQL". An automated search was conducted using these search strings using the search engines of numerous digital libraries. The paper selection criteria were then used to further filter out the most relevant research in this field. The rest of the paper discusses the database backup and recovery techniques that we derived from the review.

### *B. Test the Backup and Recovery methods of different database*

A database backup and recovery test implementation for various databases was performed in addition to the systematic review. MySQL and NoSQL database techniques are used for the test. The backup and recovery commands of MySQL and MongoDB have been used to compare the two databases

## IV. RESULT AND DISCUSSION

### *A. Full, Incremental, and Differential Backups*

Table 2: full Vs. incremental Vs. differential Backups

| Feature | Full Backup | Incremental Backup | Differential Backup |
|---|---|---|---|
| Cost | High | Low | Medium |
| Time | High | High | Low |
| Availability | High | High | High |
| Storage Space | High | Low | Low |
| Cost | High | Low | Medium |

A comparison of full, incremental, and differential backups is shown in Table 02. In order to keep multiple copies of a complete file system, the full backup factor is relatively high. The cost of the backup depends on the number of locations and the size of the files being backed up. Incremental generally copies the last changes while differential copies the updated and original file system, hence those are identified as low and medium respectively.

It takes more time to do a full or incremental backup than the difference between the two. It's because of the most recent incremental upgrade and full file backup. However, there is no such problem as differential backup.

The backups' availability features can be accessed round-the-clock. As a result, all backups are readily available and highly efficient. Whenever a user needs data, they can get it instantly and without error.

Full backup uses a lot more space than the other two because it copies the complete file system. In contrast, the remaining two perform a small number of file backups

### *B. Backup techniques, databases, and other features*

Table 3: Backup and Recovery Technique and the findings

| Author(s) | Technique | Database | Findings |
|---|---|---|---|
| Karina Bohora, et al. | Typical Backup and Recovery | Cassandra NoSQL | Failure identification and management |
| Hwajung Kim et al. | log data for write-ahead logging (WAL), | MySQL | crash recovery procedure of WAL, throughput, latency |
| Yongseok Son et al. | SSD-assisted backup/recovery, BR-SSDs | PostgreSQL and MongoDB | evaluated database backup and recovery with different workloads, backup scenarios, and environments |
| R. Cegiela | TAIL LOG | MySQL | latest possible transaction |
| Many studies with many authors | ext3cow, BTRFS, copy-on-write (CoW), flash translation layer (FTL), SSD, hard disk drives (HDD) | MySQL, NoSQL | the storage stack, run-time performance |
| Hwajung Kim et al. | Potential Techniques | MySQL | storage device performance |

Table 03 demonstrates the diverse methodologies and conclusions of different authors. A variety of methods were employed by the authors, as can be seen in table 3. The WAL, SSD, TAIL LOG, CoW, FTL, and HDD are some of the backup and recovery techniques. The authors also discovered malfunction recognition, disaster recovery, metrics of throughput and latency, backup and recovery, workload management, transactions, storage management, and run-time performance


Faculty of Technology, South Eastern University of Sri Lanka (ISBN 978-624-5736-40-9)

### C. *Test Implementation of backup and recovery technique for MySQL and NoSQL*

#### 1) *MySQL Implementation*

The MySQL database's backup and recovery test implementation is depicted in Figure 03. The newly formed database schema is called "student," as can be deduced from figure 05's separate picture "01". Image 02 in Figure 03 illustrates the backup command and where the "student" backup is saved. Here the location of the student backup is "G:\Backup". In addition, once the backup command (Mysqldump) has been executed, the backup can be found in the place indicated in the figure.

"03" depicts the deletion of the database schema, whereas "04" depicts the construction of a schema with the same name, "student." The image "05" showed how the student database was being restored, and the image "06" showed the student schema tables that could be used after backup and restoration.

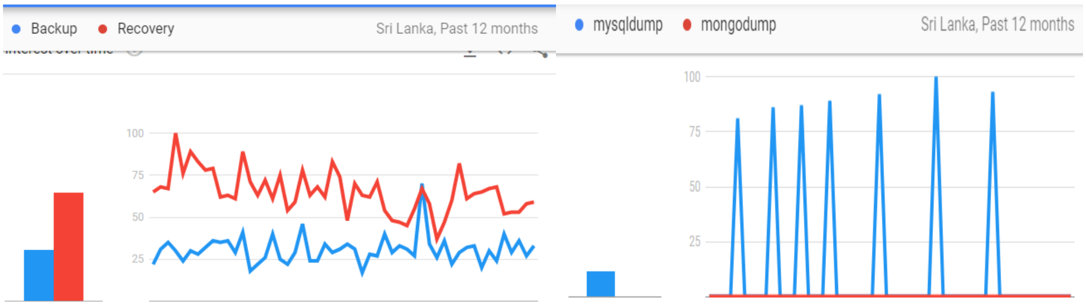


Figure 2: Google Trend search of backup, recovery, mongodump, and mysqldump

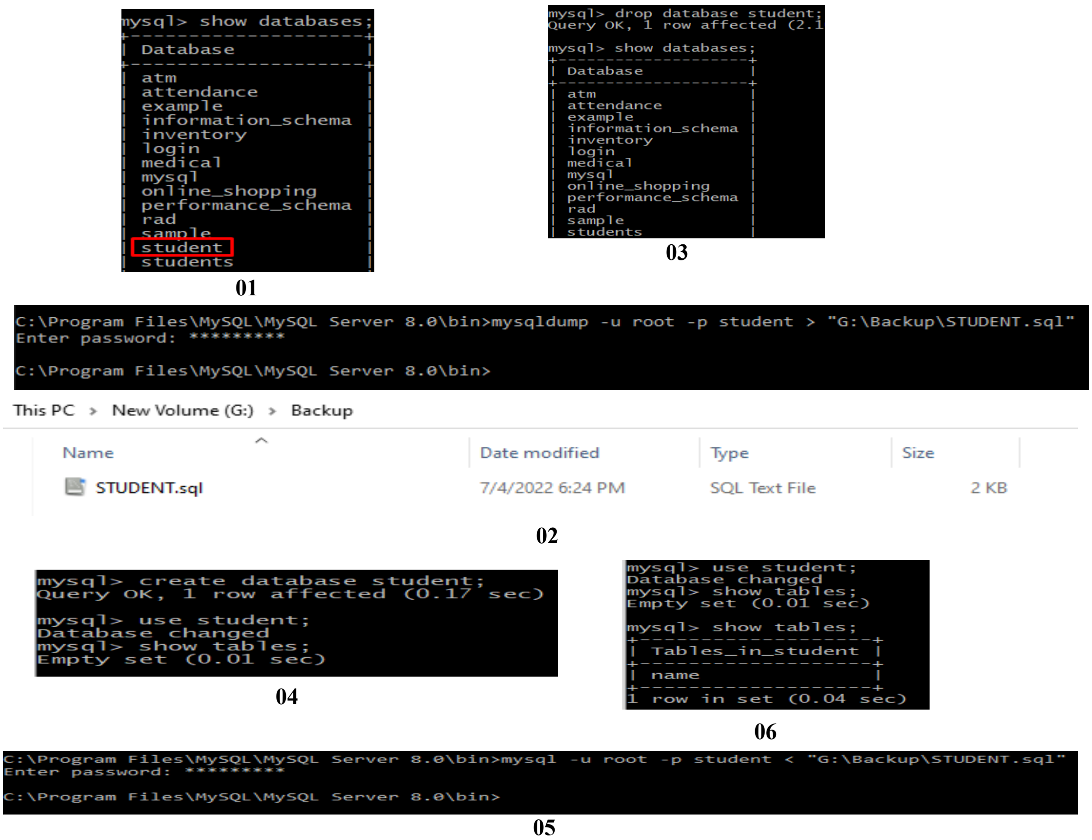


Figure 3: Backup and Recovery Process of MySQL

The test implementation allowed us to comprehend the database backup and recovery procedure. In addition, we can discover how it operates without much effort and how it gives substantial support for the MySQL database

*2) NoSQL Implementation*

Figure 04 illustrates the whole backup and recovery procedure of the NoSQL (MongoDB) database. It is evident from "01" that the newly formed database instance is entitled "myNewDB".

The image "02" in figure 06 demonstrates the wiped totally. Image “04” shows the restoration operations of the NoSQL database. Once we apply that, we can get the database schema back as in the image “05”. MySQL and NoSQL provide vast support and easy service to the backup and recovery process. The Google trends show that in the past 12 months, web search volume for recovery is very high when compared with the backup process. Furthermore, another noteworthy fact to notice is that MySQL backup is a more frequently searched backup process than NoSQL backup. From what it seems, most companies and organizations still rely on the MySQL database.

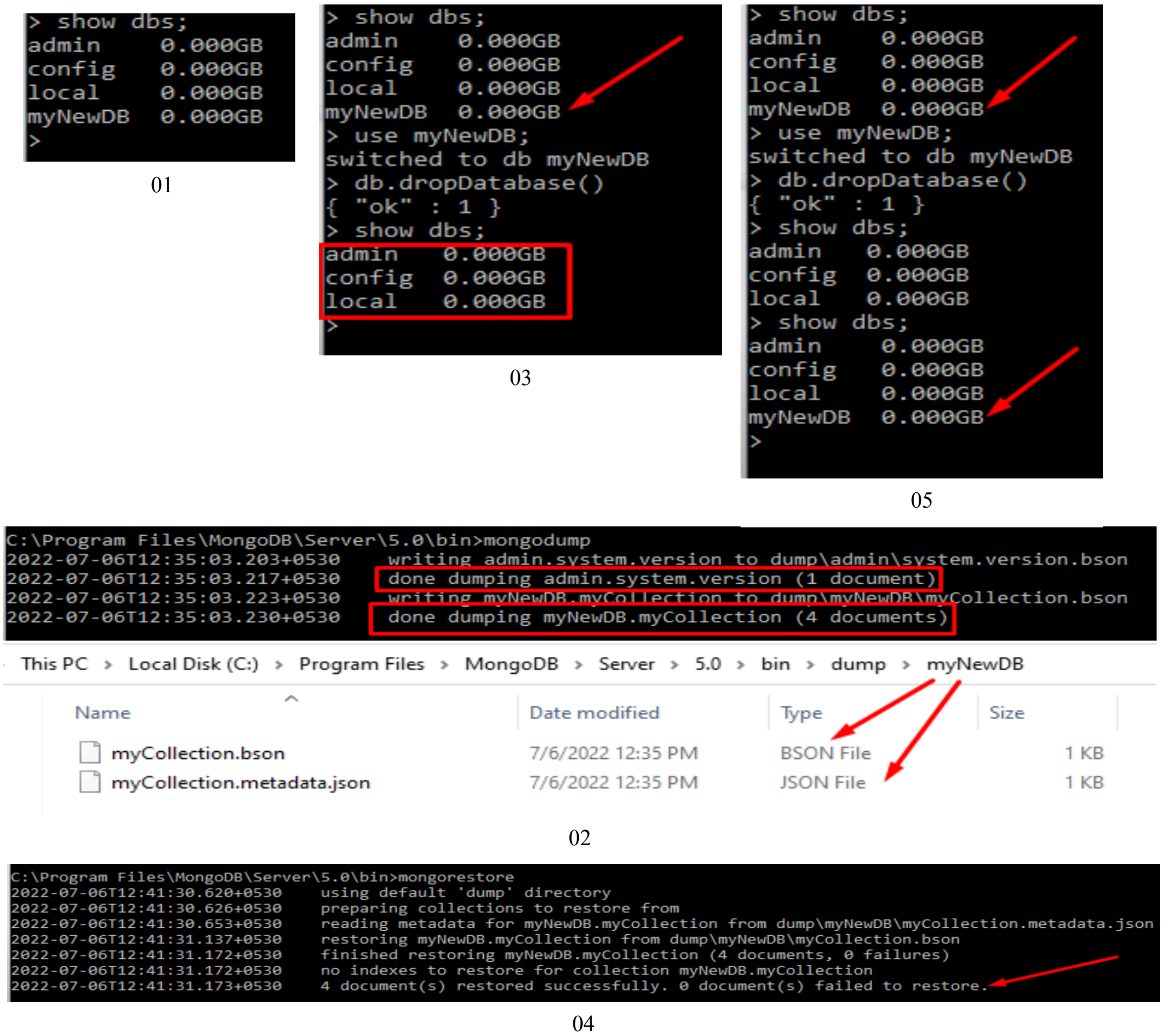


Figure 4: Backup and Recovery Process of NoSQL

NoSQL database backup procedure with the "Mongodump" command. Once we use that command, the schema is placed in the right spot. The image "03" displays the database deleting process. Once, after performing that query, we can observe that the "myNewDB" schema has been

## V. CONCLUSION

In this study, we propose efficient backup and recovery techniques for database systems with reference to the extant literature available and test implementations. Some solutions for recovering and backing up large databases have been suggested. The identified different types of database backup and recovery can be listed as full backup, differential backup, incremental backup, Log Based Recovery, and Shadow Paging. This study was able to identify the major and trending backup and recovery techniques of database systems of different databases. Most organizations do not aware of how to handle database backup and recovery. Different types of databases have a different database backup and recovery techniques. Understanding the type and working procedure of the database's backup and recovery techniques will help the organization secure the sensitive data during the calamities and will enable them to continue the service without any interruption. If the user applies the right technique and implements it correctly, this will be one of the potential remedies for database backup and recovery problems.

REFERENCES

Aleryani, A. (no date) 'The Database Administrator's Readiness to Migrate Database in the Cloud'.

Baek, S. et al. (2020) 'SSD-assisted ransomware detection and data recovery techniques', *IEEE Transactions on Computers. IEEE*, 70(10), pp. 1762–1776.

Banecs, V., Babarada, F. and Ravariu, C. (2020) 'Windows Server Backup and Restore for Moodle E-Learning Platform', *in 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–4.

Bao, L. et al. (2022) 'XML2HBase: Storing and querying large collections of XML documents using a NoSQL database system', *Journal of Parallel and Distributed Computing. Elsevier*, 161, pp. 83–99.

Bohora, K. et al. (2021) 'Backup and Recovery Mechanisms of Cassandra Database: A Review', *Journal of Digital Forensics, Security and Law*, 15(2), p. 5.

Cameselle, R. V. and Labrador, H. G. (2021) 'Addressing a billion-entries multi-petabyte distributed file system backup problem with cback: from files to objects', *in EPJ Web of Conferences*, p. 2071.

Cegiela, R. (2006) 'Selecting technology for disaster recovery', *in 2006 International Conference on Dependability of Computer Systems*, pp. 160–167.

Chang, D. et al. (2021) 'Cloud computing storage backup and recovery strategy based on secure IoT and spark', *Mobile Information Systems. Hindawi*, 2021.

Choi, M.-S. et al. (2000) 'Two-step backup mechanism for real-time main memory database recovery', *in Proceedings Seventh International Conference on Real-Time Computing Systems and Applications*, pp. 453–457.

Dagnaw, G., Hua, W. and Zhou, K. (2020) 'SSD Assisted Caching for Restore Optimization in Distributed Deduplication Environment', *in 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD\&IS)*, pp. 1–8.

Gotseva, D., Gancheva, V. and Georgiev, I. (2011) 'DATABASE BACKUP STRATEGIES AND RECOVERY MODELS', *Challenges in Higher Education & Research*, 9, pp. 147–150.

Heidari, A. and Navimipour, N. J. (2021) 'Service discovery mechanisms in cloud computing: a comprehensive and systematic literature review', *Kybernetes. Emerald Publishing Limited*.

Hu, B. (2022) 'The Application of Computer Database System in Educational Information Management', *in International Conference on Forthcoming Networks and Sustainability in the IoT Era*, pp. 141–148.

Huang, P. et al. (2012) 'BVSSD: Build built-in versioning flash-based solid state drives', *in Proceedings of the 5th Annual International Systems and Storage Conference*, pp. 1–12.

Jiang, T. et al. (2021) 'ReliableBox: Secure and Verifiable Cloud Storage With Location-Aware Backup', *IEEE Transactions on Parallel and Distributed Systems. IEEE*, 32(12), pp. 2996–3010.

Jing, F. et al. (2021) 'Remote Sensing Data Backup and Recovery System for Power Industry', *in 2021 IEEE Intl Conf on Parallel \& Distributed Processing with Applications, Big Data \& Cloud Computing, Sustainable Computing \& Communications, Social Computing \& Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 1717–1723.

Jung, Y. and others (2019) 'SSD-Insider++: SSD-Assisted RaMasteromware Defense and Data Recovery', DGIST.

Kadry, S. et al. (2011) 'A New Technique to Backup and Restore DBMS using XML and. NET Technologies', arXiv preprint arXiv:1111.5639.

Khan, M. S. A. et al. (2021) 'An analytical approach to real-time cloud services on IoT-based applications for smart city planning', *International Journal of Grid and Utility Computing.* Inderscience Publishers (IEL), 12(5–6), pp. 507–523.

Kim, H., Yeom, H. Y. and Son, Y. (2020) 'An efficient database backup and recovery scheme using write-ahead logging', *in 2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, pp. 405–413.

Ko, S.-J. et al. (2021) 'Common Data Model and Database System Development for the Korea Biobank Network', *Applied Sciences. MDPI*, 11(24), p. 11825.

Li, G. et al. (2021) 'opengauss: An autonomous database system', *Proceedings of the VLDB Endowment. VLDB Endowment*, 14(12), pp. 3028–3042.

Li, Q. and Xu, H. (2009) 'Research on the backup mechanism of oracle database', *in 2009 International Conference on Environmental Science and Information Application Technology*, pp. 423–426.

Maher, R. and Nasr, O. A. (2021) 'DropStore: A secure backup system using multi-cloud and fog computing', *IEEE Access*. IEEE, 9, pp. 71318–71327.

Makris, A. et al. (2019) 'Performance Evaluation of MongoDB and PostgreSQL for Spatio-temporal Data.', *in EDBT/ICDT Workshops*.

Makris, A. et al. (2021) 'MongoDB Vs PostgreSQL: A comparative study on performance aspects', *GeoInformatica.* Springer, 25(2), pp. 243–268.

Mishra, S., Sharma, S. K. and Alowaidi, M. A. (2021) 'Analysis of security issues of cloud-based web applications', *Journal of Ambient Intelligence and Humanized Computing. Springer*, 12(7), pp. 7051–7062.

Peterson, Z. and Burns, R. (2005) 'Ext3cow: A time-shifting file system for regulatory compliance', *ACM Transactions on Storage (TOS). ACM New York, NY, USA*, 1(2), pp. 190–212.

Rodeh, O., Bacik, J. and Mason, C. (2013) 'BTRFS: The Linux B-tree filesystem', *ACM Transactions on Storage (TOS). ACM New York, NY, USA*, 9(3), pp. 1–32.

Sahu, P., Raghavan, S. and Chandrasekaran, K. (2021) 'Ensemble deep neural network based quality of service prediction for cloud service recommendation', *Neurocomputing. Elsevier*, 465, pp. 476–489.

Sharma, M., Sharma, V. D. and Bundele, M. M. (2018) 'Performance analysis of RDBMS and no SQL databases: PostgreSQL, MongoDB and Neo4j', *in 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–5.

Sharma, S. et al. (2012) 'Analysis of recovery techniques in data base management system', *Research Journal of Computer and Information Technology Sciences*, 2320, p. 6527.

Son, Y. et al. (2017) 'SSD-assisted backup and recovery for database systems', *in 2017 IEEE 33rd International Conference on Data Engineering (ICDE*), pp. 285–296.

Woltmann, L. et al. (2021) 'PostCENN: postgresql with machine learning models for cardinality estimation', *Proceedings of the VLDB Endowment. VLDB Endowment,* 14(12), pp. 2715–2718.

Zhang, Y. et al. (2022) 'Distributed data backup and recovery for software-defined wide area network controllers', *Transactions on Emerging Telecommunications Technologies. Wiley Online Library*, 33(4), p. e4411.

Zhang, Y., Xu, C. and Muntean, G.--M. (2021) 'A Novel Distributed Data Backup and Recovery Method for Software Defined-WAN Controllers', *in 2021 IEEE Global Communications Conference (GLOBECOM*), pp. 1–6.

Zimányi, E., Sakr, M. and Lesuisse, A. (2020) 'MobilityDB: A mobility database based on PostgreSQL and PostGIS', *ACM Transactions on Database Systems (TODS)*. ACM New York, NY, USA, 45(4), pp. 1–42.