International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 5, Issue 1, 83-91, 2024
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V5I1P110

Original Article

Migrating Databases: Best Practices For Upgrading Legacy Applications

Arun K Gangula
Independent Researcher.

Abstract *- The research examines database migration as a crucial process for upgrading outdated applications through a comprehensive evaluation. The paper explains essential terms while discussing the reasons behind complex projects and describes both the various obstacles and potential dangers that exist. The report utilizes existing literature and industry insights to present multiple migration strategies, focusing on best practices for pre-migration planning and execution, as well as post-migration validation phases. The research provides IT architects and decision-makers with a comprehensive framework for executing database migrations, protecting data integrity, and minimizing system disruptions to achieve the full benefits of a modern system.*

Keywords *- Database Migration, Legacy Systems, Application Modernization, Data Management, Best Practices, Data Quality, Downtime, Risk Management, Migration Strategies.*

1. Introduction

The official process of database migration involves transferring data from a source database to a target environment, enabling the complete decommissioning of the source system [1]. The process serves as a fundamental requirement for legacy application modernization because it addresses the operational challenges of basic systems built with outdated technology [2]. These legacy systems remain operational because their replacement incurs high costs, along with complex implementation and perceived security risks. The modernization process addresses these challenges by either upgrading or replacing these systems to deliver improved functionality and platform compatibility, while reducing their substantial maintenance requirements. The deep connection between applications and their database infrastructure makes database migration projects essential for every major modernization initiative.

An adequately designed migration strategy holds essential strategic value. Organizations that want to stay competitive while adapting to changing business needs must perform this critical initiative. Organizations face significant financial and operational risks when they lack proper migration strategies because this leads to extreme budget increases, delayed project timelines, and complete project abandonment [3]. The fast-moving technological environment renders standing still an unsustainable long-term approach, as all systems will eventually create barriers to advancement. The decision to migrate represents a required business choice rather than an optional technical enhancement. The process stands as an unavoidable core element of contemporary IT administration, which ensures organizations maintain their digital relevance and growth potential [4].

2. Understanding Legacy Systems and Database Migration

2.1. Defining Legacy Applications: Characteristics and Impact

Legacy applications comprise outdated software, hardware systems, and programming environments that organizations maintain despite their non-compliance with contemporary technology standards and business requirements. Common characteristics include:

- **Outdated Technology:** The systems use outdated programming languages, including COBOL and FORTRAN, together with legacy mainframe hardware [2].
- **Inflexibility:** The systems maintain inflexible architectures that prevent them from adapting to new requirements and integrating with contemporary tools.
- **High Maintenance Costs:** The systems need expensive maintenance because vendors provide minimal support, and few qualified experts can work on them.
- **Security Vulnerabilities:** Legacy systems expose themselves to breaches because they do not contain modern security features [5].
- **Poor Efficiency and Scalability:** These systems operate at slow speeds and fail to expand when data volumes increase or user numbers grow.

- **Compliance Risks:** The systems do not comply with GDPR and other updated regulations, which increases the chances of non-compliance.
- **Data Silos:** The systems operate as separate islands, which prevent staff across the organization from accessing information.
- **Dependence on Individuals:** The systems' operational risks stem from the fact that essential knowledge about them exists only with a small group of staff members.
- **Lack of Documentation:** The scarce and outdated documentation makes it challenging to maintain and upgrade the systems.

The continued use of legacy systems creates scalability limitations, blocks innovation, and accumulates technical debt, resulting in unpredictable costs and risks. [6] The decision to avoid modernization does not yield cost savings because it leads to increased ongoing expenses, including security breaches (averaging $4M per incident), regulatory fines, and lost opportunities [5]. The investment in modernization serves to decrease these continuous financial responsibilities.

2.2. Defining Database Migration: Core Concepts and Distinctions

The process of database migration involves a structured method to move data between different environments, such as moving from on-premises to cloud platforms or from one DBMS to another (e.g., MySQL to PostgreSQL) [7]. The process of database migration includes data scrubbing, followed by schema mapping, development and testing of conversion programs, and data transfer and reconciliation.

2.2.1. Important distinctions include:

- **Migration vs. Replication:**
  - The process of migration requires a single data transfer operation, which leads to the eventual retirement of the source system [1].
  - The process of replication maintains continuous data synchronization between systems without requiring the removal of the original database.
- **Partial vs. Complete Migration:**
  - The migration process can handle complete datasets or data subsets and handle progressive modifications [1].
- **Types of Migration:**
  - The process of migration handles database structure transfers, including tables and constraints, when working with different database systems.
  - The actual records are transferred through data migration.
  - The process of application migration requires either re-platforming, rewriting, or rebuilding applications for a new system.
  - The additional strategies include storage and business process migration [8].

2.3. Motivations for Modernization and Migration

Organizations choose to migrate their technical infrastructure because it enables them to match their systems with changing business requirements [7].

The main factors include:

- Legacy systems require high expenses for infrastructure maintenance, specialized personnel, and equipment. Cloud migration enables organizations to reduce operational costs while making their systems more manageable [9].
- The adoption of data lakes through flexible systems allows businesses to gain faster insights while becoming more competitive in their markets [6].
- The new systems provide better security capabilities and assist organizations in fulfilling GDPR and other compliance standards. The migration process enables organizations to clean and enhance their data [7].
- The modern databases provide enhanced performance capabilities and scalability features, and data consolidation benefits, which become essential after mergers or when organizations break down their silos [10].

Drivers function as an interconnected system because business agility requires modern technology, and security concerns directly affect compliance and business risk [11]. Organizations need to establish a connection between technical improvements and business value through clear explanations of both objectives and reasons, ensuring that IT initiatives support strategic goals.

3. Challenges and Risks In Database Migration And Legacy Modernization

3.1. Technical Challenges in Database Migration

The process of database migration, together with legacy system modernization, presents multiple complex risks during execution. Industry data indicates that numerous projects surpass their budgets and extend beyond their timelines before ultimately collapsing [12].

Key Technical Challenges

- **Compatibility Issues:** The different DBMS systems, such as SQL Server, Oracle, MySQL, and PostgreSQL, demand extensive rewriting of queries and stored procedures because they employ unique syntax and data types and system functions [12]. Legacy systems often encounter compatibility issues with current hardware platforms, software systems, and operating system environments.
- **Data Integrity and Accuracy:** The improper execution of data mapping and transformation steps leads to data loss and corruption as well as duplication and inconsistencies [12]. The combination of hidden errors, including broken foreign keys and rounding inaccuracies, produces silent distortions in reports and analytics.
- **Performance Degradation:** The migration process leads to performance slowdowns because of incorrect index settings and variations in query optimization and unoptimized default parameters. The inefficient migration pipelines create bottlenecks, which decrease system responsiveness.
- **Application Incompatibility:** The migrated database will cause applications that depend on it to fail unless they receive proper adaptation and thorough testing. The migration process may encounter problems due to SQL syntax modifications, driver compatibility issues, and changes in performance profiles.
- **Complex Legacy Data Structures:** The complex and poorly documented legacy data models with outdated business logic create multiple errors during extraction and transformation processes [13].

The technical debt in legacy systems exacerbates these challenges because they often contain poor design elements, outdated logic, and incomplete documentation [14]. The migration of "bad data" from these systems tends to worsen data quality, increasing the likelihood of system failure in the new environment [4]. Legacy system flaws function as risk amplifiers during modernization projects, making every technical step in the migration process more complicated.

3.2. Operational Risks: Downtime, Data Loss, Security Vulnerabilities

Operational risks represent immediate dangers to business continuity, together with data integrity, both during and following migration efforts:

- **Downtime and Business Disruption:** The most significant and usually tough challenge during system migration involves reducing downtime and minimizing business disruption. [12]. System unavailability causes unplanned downtime, resulting in severe financial losses for organizations as it prevents them from accessing vital applications and data. [7] Organizations face two primary challenges: they underestimate the duration of data transfer and system reconfiguration needs, and fail to develop effective rollback and contingency strategies. [12]
- **Data Loss or Corruption:** This is a paramount risk. Data loss becomes possible because legacy data lacks proper structure or requires inadequate cleansing before the migration begins. [12] The best planning, along with execution, cannot prevent unforeseen complications that might result in the loss or corruption of important information. [15]
- **Security Vulnerabilities:** The protection of data integrity stands as a fundamental requirement for migration security purposes. [12] Organizations encounter major compliance problems and data exposure risks because they fail to properly maintain and restore all security configurations, including user roles and access policies, as well as encryption settings and audit trails, in the new system. [16] The lack of proper security management during the migration process creates the potential for new system vulnerabilities to appear.
- **Data Silos:** The difficulty of migrating data into a unified system increases because legacy systems naturally produce isolated information pockets, which become known as data silos. The fragmented nature of data creates barriers for achieving complete organizational data visibility. [5]

Data loss, along with corruption and security breaches, poses persistent dangers that compromise the reliability of business intelligence through report inaccuracies and analytics distortions, ultimately damaging stakeholder trust. These risks create immediate disruptions but result in enduring damage to both organizational decision-making abilities and stakeholder trust. The critical need for end-to-end validation and proactive monitoring exists throughout the entire migration process. The absence of clear visibility into the migration process, combined with its resulting data quality, makes it possible for "unforeseen issues" [1] and "unannounced outages" [12] to severely damage trust, thereby making it harder to obtain future IT investments and validate system integrity.

3.3. Organizational and Financial Hurdles: Resistance to Change, High Costs, Skill Gaps

Organizational and financial elements, together with technical operational obstacles, create significant difficulties for the project:

- **Resistance to Change:** A major challenge arises from human factors since employees and key decision-makers often resist changes to familiar systems. The implementation requires additional drive, combined with purposeful education and continuous guidance, to achieve successful adoption. Users generally want to maintain their existing work processes without interruption when adopting new applications; however, any perceived loss of functionality or system disruptions can lead to adverse reactions and resistance. [4]
- **High Costs:** The financial costs required for modernization projects often exceed the initial budget allocations, posing a significant challenge. The migration expenses include both migration costs and business interruption expenses, as well as unforeseen costs and the costs of diverting resources from other projects.
- **Skill Gaps:** The successful execution of database migrations requires specialized expertise that combines knowledge of source and target databases, networking, storage management, and complex application integrations. [17] The market faces a significant challenge because few professionals possess the necessary qualifications to work with outdated programming languages or technologies or to utilize advanced migration tools.
- **Lack of Documentation:** The fundamental challenge in legacy system modernization emerges from the widespread absence of complete documentation for past database migrations and the complex operations of legacy systems. [18] A shortage of institutional knowledge complicates planning and execution processes.
- **Complexity of Orchestration:** The operational risk becomes unacceptably high when attempting to retire a legacy system within a single, large-scale project that spans multiple years. The need for an incremental approach leads to new coordination problems.

Project success heavily depends on human and organizational factors, which often surpass technical obstacles as significant determinants of success. The persistent challenges of "personnel unwillingness to adjust" [17] and the "lack of skilled resources" are frequently mentioned, along with the continuous need for "training". Projects face failure risks due to their dependence on essential personnel who maintain proprietary knowledge of legacy systems. A well-structured migration plan will fail without proper change management practices, dedicated stakeholder involvement, and workforce upskilling initiatives. The human element serves as a crucial success factor that requires equal attention to technical aspects.

4. Database Migration Strategies and Approaches

The selection of appropriate database migration strategies determines essential project factors, including timeline duration, budget requirements, and system complexity. [9] The selection process must align with the organization's operational needs, risk tolerance, and resource availability.

4.1. Overview of Primary Migration Strategies

Database migration strategies include Big Bang, Trickle (also known as Phased or Incremental), and Zero-Downtime migration as their three main approaches. [9] Each strategy presents a distinct balance of advantages and disadvantages.

4.1.1. Big Bang Migration:

- **Description:** The Big Bang migration strategy involves transferring complete data from the source system to the target database in a single, large operation at a scheduled time. The migration process occurs during scheduled downtime periods, including weekends and maintenance windows. [9]
- **Advantages:** The main advantage of this approach lies in its straightforward nature, as the entire transition process occurs within a single, defined period. The consolidated nature of this approach makes it less expensive than iterative migration strategies. The correct management of this approach results in minimal operational disruptions after the cutover process is complete. [15] The elimination of dual system operation through this approach makes it easier to decommission the legacy environment. [19] The straightforward nature of this approach leads to easier testing procedures. [15]
- **Disadvantages:** The primary disadvantage of this approach is the requirement for substantial downtime, which is unacceptable for organizations that require continuous operation. [9] The strategy exposes all organizational data to a high risk of expensive failure because it conducts a single simultaneous transfer of data. The actual migration window offers limited opportunities for conducting extensive testing procedures. The short duration for flawless execution creates substantial stress for IT staff during this period. [15] Implementing a basic fallback strategy becomes challenging after the transition process begins.

4.1.2. Trickle/Phased Migration:

- **Description:** The method divides the complete migration process into smaller sections, following an agile approach. The migration process involves separate sub migrations, each with its own specific goals and specified deadlines. [9] Data migration occurs in a step-by-step process, enabling a smooth transition.
- **Advantages:** The approach provides better resistance to unexpected failures because the knowledge gained from each failed step enables improvements for upcoming phases. The approach enables verification of success during each separate phase. The old system can continue to operate at full capacity throughout the migration process, minimizing or eliminating business downtime. [9] The method maintains fresh data through continuous updates during its prolonged migration duration. The method enables controlled migration of business or data parts, which reduces total risk exposure. [19]
- **Disadvantages:** A Big Bang approach finishes the migration process faster than trickle migration methods, but the latter requires more time for completion. Running and maintaining two systems (source and target) simultaneously throughout an extended period requires additional resources and effort. The management of this strategy becomes more complex because it needs a bridge to handle data serving between the old and new systems. The implementation of this approach creates difficulties when synchronizing data between systems and managing environments that combine old and new components. [12]

4.1.3. Zero-Downtime Database Migration:

- **Description:** This advanced database migration technique enables data transfer from the source to the target databases while maintaining continuous access to the source system. The combination of advanced constant replication methods with Change Data Capture (CDC) technology enables real-time tracking and spreading of changes. [20]
- **Benefits:** The method provides minimal disruption to business operations and accelerates total migration duration while reducing downtime expenses. The strategy serves as an effective tool for reducing business impact when business continuity is the top priority for the firm. [9]

Table 1: Comparison of Database Migration Strategies

Strategy	Description	Advantages	Disadvantages	Best Use Case
Big Bang	Complete data transfer in one scheduled downtime	Simple; faster; minimal post-cutover disruption	Extended downtime; high risk; limited rollback; stressful	Small/static data; tight timelines; downtime acceptable
Trickle/Phased	Incremental migration while the old system runs	Low risk; minimal downtime; continuous updates	Longer duration, dual system overhead, complex coordination	Large/active data; low downtime tolerance; interdependent systems
Zero-Downtime	Real-time replication during active system use	No disruption; seamless cutover	High complexity; requires advanced tooling (e.g., CDC)	24/7 systems; critical uptime; zero business impact required

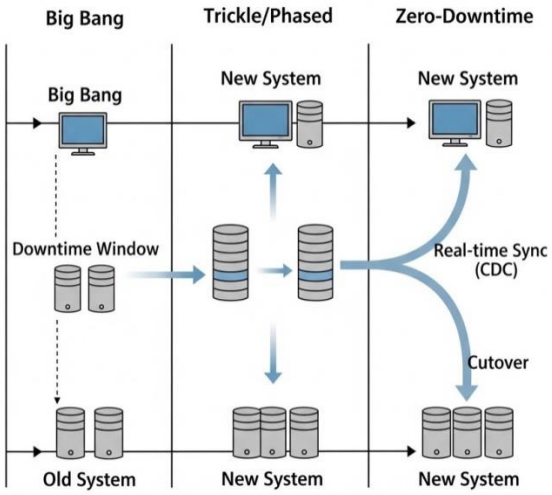

Fig 1: A Diagram with Three Parallel Timelines Illustrating the Operational State of the Source and Target Systems for Each Strategy

Companies need to evaluate the advantages and disadvantages of Big Bang and Trickle/Phased migrations before selecting their migration approach. The Big Bang method offers fast implementation but poses significant risks and necessitates extensive system downtime. The Trickle method extends the migration duration to reduce risks and downtime but demands additional resources for parallel operation execution. The zero-downtime migration approach is the most challenging method because it relies on sophisticated replication tools, including CDC, for its implementation. [9] The optimal migration approach depends on specific organizational requirements, as there is no universal best practice. Organizations must evaluate their particular risk tolerance, downtime tolerance, resource availability, and data application complexity to make informed strategic decisions

4.2. Homogeneous vs. Heterogeneous Migrations

Database migrations can be categorized into two distinct types: systems with similar database structures and systems with different database structures.

- **Homogeneous Migration:** The database migration process known as Homogeneous Migration occurs between systems that operate with the same database type or vendor (for example, Oracle to Oracle or SQL Server to SQL Server) [21]. The migration process remains straightforward because schema structures and application queries remain unchanged, which results in minimal code updates [22].
- **Heterogeneous Migration:** The process of migrating data between different database platforms is known as heterogeneous migration, which involves moving data from one platform to another, such as from MySQL to PostgreSQL, or from SQL Server to Oracle, or from relational databases to NoSQL databases. The migration process requires extensive modifications to database structures, data types, and queries due to differences in language syntax and semantics [22]. The application layer requires extensive code rewriting and testing to ensure that business logic functions correctly in the new environment during heterogeneous migration. The process introduces additional complexity, costs, and risks that exceed standard transport requirements [22].

4.3. Architectural Considerations for Migration Planning

The organization's readiness, quality standards, changing environments, and data complexity require a proper data migration architecture [7].

Key considerations include:

- **Scope Identification:** The project scope definition must include specific datasets and data items to establish realistic boundaries [23].
- **Source and Target Analysis:** The analysis of both systems should start early to understand their operations, dependencies, and potential compatibility issues [23].
- **Resource and Tool Evaluation:** The assessment of resources and tools requires an evaluation of team member skills and identification of necessary external support. The team should evaluate migration technologies based on their flexibility and their ability to match the team's skill set [23].
- **Migration Design:** The migration process requires detailed planning for extraction, followed by transformation, loading, and verification steps. Data mapping rules, along with data quality checking procedures, need to be defined clearly.
- **Recovery Plans:** The development of backup plans for each phase requires testing to ensure business continuity and rollback capabilities.
- **Deployment Schedule:** Plan migrations during times when there isn't much traffic to keep things running smoothly [23].
- **Parallel Processing Optimization:** The ETL performance can be enhanced through the implementation of multi-threading along with increased parallelism, additional loaders, and optimized table spaces and indexes [7].

Data transfer happens in steps, not in a straight line. The "Big Bang" migration approach requires multiple iterations of profiling, design, and testing before it can be implemented. The architectural design should allow for iterative refinement and staggered deployment, especially for complex systems, to maintain adaptability and continuous operation during modernization [7].

5. Best Practices for Successful Database Migration

A successful database migration requires detailed planning followed by precise execution and complete post-migration validation. The entire process depends on each phase to minimize risks and deliver long-term system value.

5.1. Pre-Migration Phase: Planning and Assessment

- **Comprehensive Data and Schema Analysis:** The analysis must examine database schema dependencies, cross-database

join operations, data formats, and data access patterns [23]. The evaluation of concealed system connections and database structures will enable organizations to stop data integrity breakdowns. The process of schema and metadata mapping demands strict monitoring to stop both formatting issues and conversion errors.

- **Defining Scope, Goals, and Resources:** The migration plan needs to specify exact timeframes, work boundaries, and resource distribution details. The assessment of internal project capabilities should be conducted to determine if external support is needed for the project execution [23] [3]. The assessment of licensing needs and feature dependencies should start at the beginning to stop unexpected problems from occurring during the last stages of development**.**
- **Data Cleansing and Quality Assurance:** The migration process needs data that contains no errors, no duplicates, and no inconsistencies. The organization needs to create data quality standards, which should begin with essential datasets, including customer records and compliance data [12]. The success rate of migration depends directly on investments toward data cleansing tools and processes [8].
- **Robust Backup and Recovery Planning:** System recovery becomes possible through the development of rollback procedures, which include testing three recovery strategies: fallback, fall forward, and dual-write [24].

5.2. Execution Phase: Implementation and Monitoring

- **Use of Appropriate Tools and Automation:** ETL and source-target optimized tools such as AWS DMS, Azure DMS, Talend, SSMA, Hevo, Informatica and Qlik Replicate should be used for this purpose [20]. The process should be automated whenever possible to achieve better accuracy and speed.
- **Maintaining Data Integrity and Consistency:** The transfer process should include schema comparisons, referential integrity checks, and checksum validation [23]. Real-time process monitoring should be implemented to prevent performance degradation on active systems while controlling resource consumption.
- **Minimizing Downtime and Disruption:** Migrations should run during periods of low system activity, while critical systems should be migrated through phased approaches, and high-volume tables should be isolated. Real-time sync and CDC, and dual-write methods should be used to achieve zero-downtime goals according to [20][24].

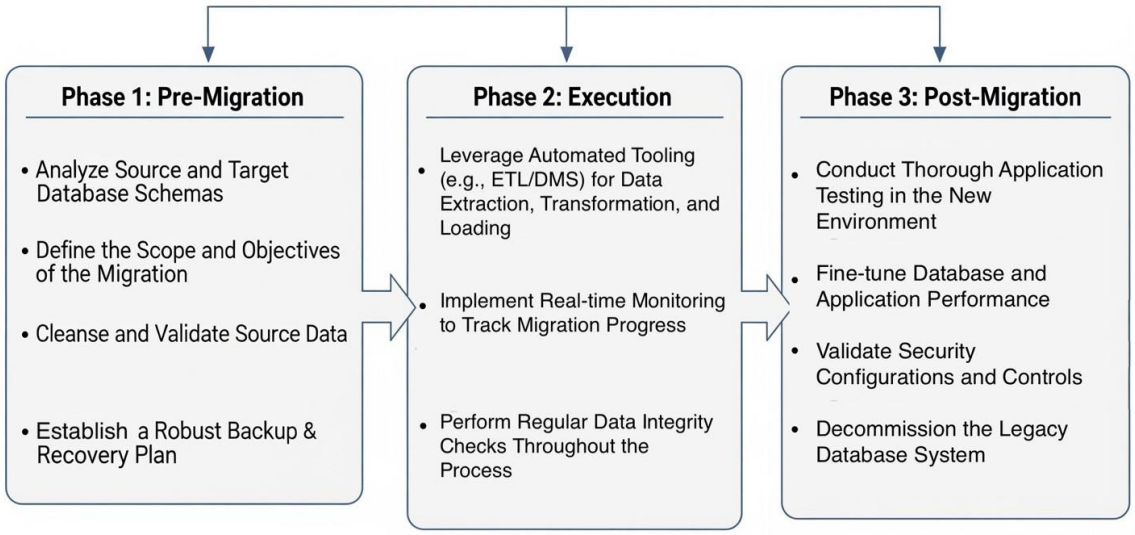

Fig 2: A high-level process flowchart that visually represents the three main phases of migration

5.3. Post-Migration Phase: Validation and Optimization

- **Thorough Data and Application Testing:** The validation process for application connectivity requires unit/system/volume tests and SQL scripts to compare data before and after migration [23]. The validation process checks data formats and types while searching for duplicate entries and access problems.
- **Performance Tuning and Monitoring:** The postmigration process requires users to compare query plans, rebuild indexes, refresh statistics, and monitor performance through dashboards [12]. The migration live operation begins the process of ongoing system enhancement.
- **Revalidating Security and Compliance:** The review process should focus on jobs, alerts, access controls, backup, and restore procedures. The implementation of replication systems and DR systems requires configuration for audit record transfer to meet compliance standards.
- **Rollback and Fallback Readiness:** The organization must maintain tested rollback scripts as part of its standard

procedure. The rollback environments need to maintain synchronization with the new system at all times [24].

- **Phasing Out Legacy Systems:** The temporary operation of legacy systems should be restricted to read-only access. The organization should remove old systems from backup systems and monitoring tools while archiving important data to stop resource consumption.

Table 2: Key Best Practices across Database Migration Phases

Migration Phase	Best Practice Area	Action/Guideline	Rationale/Benefit
Pre-Migration	Planning & Assessment	Analyze schema dependencies; perform audits	Prevents broken queries and uncovers structural issues
		Define scope, goals, budgets, and resources	Aligns project plans and ensures effective execution
	Data Quality	Clean, de-duplicate, and standardize data	Enhances accuracy and consistency
		Establish backup and recovery plans with immutable backups	Safeguards against data loss; enables quick recovery
Execution	Tools & Technology	Use optimized tools (ETL, DMS)	Increases automation, reduces errors
	Data Integrity	Run schema comparisons, validate referential integrity, use incremental checks	Ensures accurate transfer and preserves data relationships
	Downtime Minimization	Migrate during low-traffic; throttle resource usage	Minimizes business disruption and performance impact
Post-Migration	Validation	Perform data- and application-level testing	Verifies accuracy and ensures full functionality
	Performance Optimization	Tune queries, rebuild indexes, refresh stats	Improves performance and resource efficiency
	Security & Compliance	Revalidate access, audit logs, and disaster recovery	Maintains security and regulatory compliance
	Rollback & Decommissioning	Maintain rollback plan; phase out legacy systems (e.g., read-only mode)	Enables safe fallback and gradual transition

6. Conclusion

The modern IT world demands database migration as an essential yet complicated process for upgrading outdated applications. The essential nature of database migration stems from multiple essential factors, which include cost reduction needs, organizational flexibility requirements, security protection against modern threats, and complete utilization of present-day technology capabilities. The migration process contains multiple obstacles that create difficulties at every step. The process of migration faces multiple technical obstacles, which include system compatibility verification and data preservation during changes, and performance optimization challenges. System downtime, along with data loss risks and security exposure, represents major operational challenges that organizations must handle during database migration. The combination of human and financial obstacles, including employee resistance to change, high migration expenses, and insufficient workforce expertise, makes these challenges more difficult to overcome. The technical debt that accumulates in legacy systems makes every step of database migration more complicated. Database migration success requires more than just technical execution, as it involves an ongoing strategic approach to data management. The process demands that organizations maintain permanent oversight and performance optimization and develop a workplace environment that supports change initiatives and maintains data precision. Database migration requires organizations to treat it as a strategic business investment that builds their competitive advantage and operational flexibility for enduring success in an ever-changing technological and business world.

References

[1] Mudunuri L.N.R.; (December, 2023); "AI-Driven Inventory Management: Never Run Out, Never Overstock"; International Journal of Advances in Engineering Research; Vol 26, Issue 6; 24-36

[2] Praveen Kumar Maroju, "Optimizing Mortgage Loan Processing in Capital Markets: A Machine Learning Approach, " International Journal of Innovations in Scientific Engineering, 17(1), PP. 36-55 , April 2023.

[3] J. Kutay, "An introduction to database migration strategy and best practices," *Striim*. https://www.striim.com/blog/an-introduction-to-database-migration-strategy-and-best-practices/

[4] Ralwasser, "Data migration – dangerous or essential?," *Stack Exchange*, Feb. 09, 2011. https://softwareengineering.stackexchange.com/questions/50703/data-migration-dangerous-or-essential

[5] IR Team, "What are Legacy Systems and Why Do Companies Still Use Them? | IR." https://www.ir.com/guides/what-are-

legacy-systems-and-why-do-companies-still-use-them
[6] Pavel K., "Legacy System Modernization Approaches: Practical Advice on Dealing with Outdated Software," *Legacy Software Modernization Company | ModLogix*, Oct. 30, 2020. https://modlogix.com/blog/legacy-system-modernization-approaches-practical-advice-on-dealing-with-outdated-software/
[7] S. S. Sarmah, "Data migration," *Science and Technology*, vol. 8, pp. 1–10, 2018, doi: 10.5923/j.scit.20180801.01.
[8] O. Azeroual and M. Jha, "Without data quality, there is no data migration," *Big Data and Cognitive Computing*, vol. 5, no. 2, p. 24, May 2021, doi: 10.3390/bdcc5020024.
[9] Informatica, "All about database migration," *Informatica*. https://www.informatica.com/resources/articles/what-is-database-migration.html
[10] H. Dhaduk, "Database migration – what it is, understanding needs, and challenges," *Simform*, Feb. 17, 2020. https://www.simform.com/blog/database-migration/
[11] Anton V., "Risks and Approaches of Legacy Application UI and UX modernization," *Legacy Software Modernization Company | ModLogix*, Nov. 20, 2020. https://modlogix.com/blog/risks-and-approaches-of-legacy-application-ui-ux-modernization/
[12] M. Elamparithi and V. Anuratha, "A review on database migration strategies, techniques and tools," *World Journal of Computer Application and Technology*, vol. 3, no. 3, pp. 41–48, Dec. 2015, doi: 10.13189/wjcat.2015.030301.
[13] M. Srinivas, G. Ramakrishna, K. R. Rao, and E. S. Babu, "Analysis of Legacy System in Software Application Development: A Comparative survey," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 1, pp. 292–297, Feb. 2016, doi: 10.11591/ijece.v6i1.8367.
[14] M. Ansar, M. W. Ashraf, and M. Fatima, "Data Migration in Cloud: A Systematic Review," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, pp. 73–89, 2018, [Online]. Available: http://asrjetsjournal.org/
[15] "Legacy software modernization," Jan. 2017. [Online]. Available: https://www.altexsoft.com/media/2017/01/Legacy-Software-Modernization.pdf
[16] A. K. Thomer, A. J. Rayburn, and A. R. B. Tyler, "Three approaches to documenting database migrations," *International Journal of Digital Curation*, vol. 15, no. 1, p. 5, Aug. 2020, doi: 10.2218/ijdc.v15i1.726.
[17] D. Jones, "Data Migration Test Strategy: Create an effective test plan — Data Migration Pro," *Data Migration Pro*. https://www.datamigrationpro.com/data-migration-go-live-strategy
[18] "7 Steps to Include in your Data Migration Plan | Secure Cloud Backup Software | Nordic Backup," *Secure Cloud Backup Software | Nordic Backup*. https://nordic-backup.com/blog/7-steps-data-migration-plan/
[19] "Rolling back from a migration with AWS DMS | Amazon Web Services," *Amazon Web Services*, Apr. 22, 2020. https://aws.amazon.com/blogs/database/rolling-back-from-a-migration-with-aws-dms/