



# Proven Best Practices for Database Migration Success

# Table of Contents

An illustration of a person in a red shirt and grey pants running through a landscape of stylized database cylinders (orange and blue) and white clouds. The person is running towards the right, with a trail of small blue and orange dots behind them. The background is a light blue sky with white clouds.

5	Step one: Planning
6	Resource management: Balancing people, technology, time, and budget
6	Data preparation
7	Database setup
7	Schema and data migration
8	Migration of apps
9	Quality assurance
10	Finalize the migration
10	Documentation and evaluation
11	Get expert database migration support from Percona
12	Checklist for database migration



## Proven Best Practices for Database Migration Success

When we hear the term “database migration,” we may immediately think only of transferring data from one database environment to another. But this process is not just about moving data; it encompasses not only transferring data but also database objects, schemas, indexes, and associated applications. Various factors can prompt database migrations, such as better performance, improved scalability, adoption of multi-cloud strategies, keeping up with technological advancements, reducing costs, or breaking free from vendor lock-in.

With 60% of companies expected to move to the cloud by 2025 ([McKinsey & Company](#)) and the continuous evolution of database management systems like MySQL, PostgreSQL, and MongoDB, successful migrations have become critical for ensuring seamless, efficient, available, and secure data management.

When preparing for a migration, one can face various scenarios: upgrading to newer database versions, moving from on-premise databases to cloud platforms, or transitioning between entirely different DBMS. Regardless, each requires a meticulous approach to ensure data integrity, minimize downtime, and align with an organization’s strategic objectives.

We’ve assembled this guide to the best practices for database migration to help streamline this complex process, providing you with a framework and actionable insights to navigate each stage effectively, from initial planning to post-migration optimization and monitoring. Let’s get started.



## Step one: Planning

### Assessment: Defining goals and objectives

Initiating your database migration starts with a thorough evaluation. Begin by pinpointing the motivations behind the move. Is it to enhance performance, expand scalability, cut down on expenses, or gain access to features you currently don't have? Clearly defining these driving factors is crucial for maintaining a focused and objective-aligned migration path.

### Software selection: Finding the right fit

Choosing a suitable database that meets your needs is a critical step. Consider your specific data needs and workload to determine whether a relational (SQL) or non-relational (NoSQL) database is the better choice for your situation. Various databases, including MySQL, PostgreSQL, MongoDB, and MariaDB, offer unique benefits and have certain limitations. It's essential to assess these options based on factors like scalability, compatibility with your current systems, performance efficiency, security capabilities, the types of data they support, licensing rules, compliance with regulatory standards, and the potential learning curve of adopting a new system.

### Mapping out your migration strategy

Implementing a well-defined strategy for your migration can significantly streamline the process. This strategy must align with your objectives, capabilities, and your organization's specific needs. As you progress, there are several standard migration approaches to consider, each tailored to facilitate a smoother transition.

- *Parallel run: Running the old and new database systems side by side during the transition period. This strategy allows for a smooth switch in case any issues arise.*
- *Phased migration: Incrementally transferring select data, components, or features from the existing database to the new one. While this approach reduces operational disruptions, it typically prolongs the migration duration.*
- *Full cutover: Switching entirely to the new database in one go at a predetermined moment. This method simplifies the process but demands thorough confidence in your preparation, testing, and capabilities.*



# Resource management: Balancing people, technology, time, and budget

A successful database migration hinges on the strategic management and understanding of your resources. A primary concern revolves around your team: assessing whether you have the right individuals with the necessary skills and knowledge for this transition is crucial. Consider if additional training is needed for your current team or if you need to enlist external specialists for assistance.

Another critical component of resource planning involves evaluating your hardware needs. Examine your existing infrastructure, including hardware, storage, and networks, to ascertain compatibility with the new database. Upgrades or new hardware investments might be essential to guarantee efficient performance after the migration.

Next, it's important to establish a practical timeline for completing your migration. Consider the possibility of disruptions that could affect operations — and plan for unexpected issues.

And finally, consider your budget. In planning your database migration, ensure that you allocate sufficient funds to cover your staffing requirements (whether in-house or external), any necessary hardware modifica-

## Data preparation

### Backup

Data preparation is an essential stage in any database migration project, and the success of a transition hinges on two crucial tasks: creating a full backup of your current database and undertaking data cleanup to tailor your existing data to the new database's standards.

Executing a complete backup of your existing database is imperative. It's not optional or a task to be postponed — it's an absolute necessity. With a thorough backup securely stored, you can effectively address and manage any unforeseen complications or data loss during the migration, thereby reducing potential disruptions to your organization.

### Cleanup

With your data securely backed up (which it certainly should be), the next step is concentrating on data cleanup.

This process is vital for a successful migration and involves various tasks like converting data types and modifying schemas to improve compatibility. Different databases typically have distinct data structures, formats, and types that may not correspond with your current setup. Data cleanup involves modifying your existing data to meet these new standards, ensuring a smooth integration while preserving its integrity.

**Converting data types:** Ensure that data types in your current database are converted to those compatible with the new database, thereby preventing data-type mismatches.

**Schema modifications:** Adjust your database schema to meet the new system's requirements, which may involve renaming tables, columns, and other elements.

**Transforming data:** Modify the data (such as modifying data values and reformatting dates) to suit the structure and format of the new database.

**Data cleanup:** This is also a great time to clean out the data. Over time, databases accumulate redundant or outdated data, and starting your migration with a clean dataset simplifies it. It comes with the bonus of improved query performance in your new environment.



# Database setup

Starting your migration begins with setting up the new database software on your server or choosing a cloud-based platform. After deciding on your preferred platform, begin the installation by downloading it and following the setup instructions provided by the vendor or those commonly discussed within its user community.

Once installed, the next step is to tailor the software to your specific needs. This stage involves fine-tuning elements such as authentication methods, storage pathways, security protocols, and memory allocation to suit your particular use cases. Customizing these settings now will help you fully leverage the advantages of your new database system and lay the groundwork for smooth data migration.

## Schema and data migration

It's getting to be go-time around here! Just a few more essential steps to follow in your migration journey...

### Analyzing the schema

Before proceeding with the data migration, conducting a thorough analysis of the current schema is essential. This analysis will inform the development of a corresponding schema in the new database environment you've selected. Mapping tables, indexes, and constraints to their equivalents in the new setting is critical to preserving the relational structures and ensuring the integrity of your data throughout the migration process.

### Data migration

With your schema now established, the next step is to transfer the actual data. The tools you'll use for this task depend on your target database. Utilities like `mysqldump` for MySQL and `pg_dump` for PostgreSQL are excellent for efficiently extracting and transforming data, facilitating a smoother process and integration into the new database environment.



# Migration of apps

Having completed the data migration, the next step involves making necessary updates. Connection strings and configuration files bridge your application and the database, dictating how they communicate. Updating these components is vital when transitioning to a new database, as it helps minimize potential disruptions and maintain data consistency. In addition, switching to a different database system frequently necessitates adjustments in the code to guarantee smooth compatibility.

## Update connection strings and configuration files

**Catalog current connections:** Start by cataloging all existing connection strings or configuration files in your applications that link to the old database system.

**Configuration for new database:** Modify your connection strings and configuration files to match the new database's details, including server address, port, and authentication credentials.

**Rigorous connection testing:** Ensure your application can connect to the new database and perform data retrieval, organization, and processing as required.

**Backup and rollback planning:** Before applying these changes in a live setting, set up a backup and rollback plan to revert to the old configuration if necessary, preventing data loss.

## Modifying the code

**Align data types:** Ensure that the code, particularly those parts dependent on specific data types, is compatible with the data types of the new database.

**Update stored procedures and functions:** Adapt any stored procedures or user-defined functions used by your application for compatibility with the new database, keeping in mind that their logic and syntax can vary across different database systems.

**Ensure SQL compatibility:** Review and adjust your existing SQL queries to match the syntax and supported functions of the new database system.

**Modify security configurations:** Adjust user roles and permissions as necessary to align your application's security model with the security settings of the new database.

**Revise indexes and constraints:** Check and update any indexes, constraints, or triggers in your code to comply with the indexing rules and constraint enforcement specific to the new database.







## Quality assurance

Here, we explore crucial factors that help to ensure data integrity, maintain application functionality, and optimize performance throughout your database migration process:

### Testing

Quality assurance kicks off with detailed unit testing to confirm each component's functionality within your system. Following this, integration testing is conducted to ensure that these components interact successfully, and user acceptance testing is carried out to ensure that the system operates as anticipated from the users' perspective. By detecting and resolving potential issues at an early stage, you can effectively prevent them from escalating into more significant problems down the line.

### Performance tuning

The initial key step in performance tuning involves optimizing your queries. This crucial action aims to boost the efficiency of your database queries, thereby shortening response times for users. Additionally, creating appropriate indexes is instrumental in speeding up data retrieval, ensuring sustained peak performance as your database expands.

### Documentation

As modifications are made to your database during the migration process, keeping your documentation updated is imperative. This should include comprehensive details of changes made to the database schema, revisions in query structures, and any configuration tweaks. Such thorough documentation will aid your team in understanding the new environment and effectively addressing any issues.



## Finalize the migration

These final steps in the migration process are designed to facilitate a successful transition to your new database.

Conducting a **final data synchronization** is imperative before fully transitioning to your new database, as your data may have undergone various changes or updates throughout the migration process. This synchronization step guarantees that all such modifications and new additions are accurately incorporated into the new database and as a critical quality control measure to maintain data integrity and consistency. Completing this final data sync ensures your new database contains the latest and most accurate information.

Is everything functioning perfectly with the synchronized data in place? Great! Now, it's time to **retire your old database**. This step includes shutting down the old system and terminating all its services, connections, and processes. This step is crucial to prevent accidental data entries into the old database.

## Documentation and evaluation

As you conclude your database migration, it's important to focus on two critical aspects: thorough documentation and comprehensive evaluation. These post-migration steps are crucial in securing the enduring success of your migration project and laying the groundwork for ongoing improvements.

### Detailed documentation

Creating detailed documentation is an essential part of the post-migration phase. It encompasses more than just the migration's technical procedures; instead, it captures the project's full scope. This documentation should contain in-depth accounts of the challenges, solutions implemented, and best practices identified throughout the migration. Such a resource becomes invaluable for your IT team, aiding in troubleshooting and routine maintenance.

### Assessment and continuous improvement

Reflect on whether the migration met your objectives, like improved performance, heightened security, or reduced costs, and pinpoint any unforeseen obstacles encountered along the way. This evaluation will help you fine-tune your approach to migration for any future requirements.



# Get expert database migration support from Percona

If you find yourself needing assistance with your migration, Percona can help. Our experts in database solutions offer a range of services and support that can be invaluable for any organization looking to migrate its databases. Choosing a trusted partner can significantly simplify the migration by providing guidance, automation, and solutions to commonly faced migration challenges. Here's why Percona is the perfect choice for your database migration needs:

**Database expertise:** Percona specializes in open source and source-available databases like MySQL, PostgreSQL, and MongoDB. This expertise is crucial for businesses relying on these platforms for migration projects.

**Customized migration strategies:** Understanding that each migration project has unique requirements, Percona offers customized migration strategies. We work closely with clients to understand their needs, challenges, and objectives, ensuring a migration plan aligns with business goals.

**Performance optimization:** One of Percona's key strengths is our focus on performance optimization. During the migration process, we can provide insights and adjustments to ensure the new database environment is functional and optimized for performance.

**Comprehensive migration support:** Percona offers end-to-end support for database migrations, including pre-migration planning, data assessment, execution of the migration, and post-migration optimization. Our hands-on approach ensures that clients have expert guidance at every step.

**Risk mitigation:** Percona is well-equipped to identify potential risks associated with database migration and propose effective mitigation strategies. This proactive approach helps minimize downtime and ensure data integrity.

**Training and knowledge transfer:** Post-migration, Percona can provide training and documentation to ensure your team is well-equipped to manage the new database environment. This knowledge transfer is crucial for the long-term success of the migration project.

**24/7 support and troubleshooting:** Recognizing the critical nature of database systems, Percona offers round-the-clock support. Our team can assist with any post-migration issues or troubleshooting, ensuring that problems are swiftly resolved.

**Tools and resources:** Percona also offers a suite of tools and resources that can aid in various stages of the migration process. These tools are designed to enhance efficiency, reduce risks, and provide insights into database performance.

Choosing Percona for your database migration needs means partnering with a team with a deep understanding of database technologies, performance optimization, and the complexities of migration projects. Our expertise can be pivotal in ensuring a smooth, efficient, and successful migration.

[Contact us for a no-obligation migration assessment](#)



## Checklist for database migration

Use this thorough checklist to ensure that all critical aspects of a database migration process are covered effectively.

## Pre-migration

Define migration goals and objectives: Clearly outline the purpose and objectives for the migration.

Conduct a thorough needs analysis: Evaluate your current and future requirements, limitations, and expected results from the migration.

Choose suitable migration tools: Identify the most suitable ones, considering their compatibility with your systems, performance, user-friendliness, and vendor support.

Perform risk assessment: Spot potential challenges and create plans to mitigate these risks.

Develop a detailed migration plan: Craft a comprehensive migration strategy outlining timelines, allocation of resources, assignment of roles and responsibilities, and a plan for communication.

Prepare the source and target environments: Prepare both the source and destination databases, ensuring they are fully configured and migration-ready.

## Data preparation

Audit the existing data: Assess the data quality, consistency, and integrity in the database.

Data cleansing and de-duplication: Conduct thorough data cleaning and eliminate duplicate records to maintain data quality.

Conduct schema mapping and conversion: Align the schema of the source database with that of the target database through careful mapping and conversion.

Plan for large data volume handling: Develop strategies for managing and transferring large volumes of data.

## Testing

Set up a test environment: Establish a testing environment replicating the production setup.

Validate data and functionality: Verify the completeness and accuracy of the migrated data and confirm that all database functionalities are operating correctly.

Perform performance testing: Assess the database's performance under different load conditions to ensure it meets expected standards.

Security and compliance checks: Check that the migrated database adheres to all required security standards and compliance regulations.

## Migration execution

Initiate data transfer: Begin the process of transferring data from the source to the target database.

Monitor the migration process: Continuously oversee the migration process to identify and resolve any issues or slowdowns.

Troubleshoot issues promptly: Quickly tackle any challenges or complications during the migration.

Stakeholder engagement: Maintain regular communication with all involved parties, keeping them updated and engaged throughout the migration process.



## Post-migration

Verify data integrity and consistency: Confirm that all data has been accurately transferred and maintains consistency across the system.

Performance optimization: Adjust the new database settings to enhance performance based on the metrics observed during migration.

Update documentation: Revise technical documentation to reflect changes and new configurations.

User and staff training: Offer necessary training and resources to users and staff for adapting to the new system.

Implement post-migration review: Perform a comprehensive review of the migration process to pinpoint successes and areas for future improvements.

Establish ongoing monitoring and maintenance: Establish a systematic approach for regular monitoring and ongoing maintenance of the new database infrastructure.



**PERCONA**

[percona.com](https://percona.com)