# Database Migration

*by Sophia*

WHAT'S COVERED

In this lesson, you will learn about data migration planning. You will learn the importance of drivers to ensure that data is backed up to prevent loss and corruption during the migration processes. Specifically, this lesson will cover:



## 1. Make a Data Migration Plan

**Data migration** involves transferring data from one database to another while ensuring the data's integrity, accuracy, and consistency. There are typically a number of steps involved in a migration, including:

1. Assessing the source and target databases.
2. Planning the migration strategy.
3. Profiling and cleansing data.
4. Mapping and transforming data between the two systems.
5. Backing up the source database.
6. Conducting test migrations.
7. Executing the final migration.
8. Validating the data.

A proper migration plan and documentation are essential for ensuring a smooth and successful transition between two databases in the event of any unexpected issues.
Migrating data to a new system successfully requires a well-executed plan to ensure data integrity, minimize disruption, and set the foundation for success. A data migration plan outlines the process of transferring data between two systems, ensuring that the transition will be seamless and successful. Although the steps in a data

migration plan may vary according to the project's complexity and migration requirements, the following are some common ones:

STEP BY STEP

1. Assess and plan the source and target systems, data structure, data quality, and compatibility.
2. Identify the migration objectives, scope, timeline, and resources required. Identify the challenges and risks that may arise.
3. Profile the data to determine its structure, relationships, and quality. Then clean the data, if necessary, to eliminate any inconsistencies or anomalies that might affect the migration. For example, you might correct typos to avoid having multiple versions of what should be the same entry between records. By identifying and resolving data inconsistencies, missing values, and other issues that may affect the accuracy and reliability of the migrated data, you help ensure the data is accurate and reliable.
4. Develop a mapping document containing mapping rules that describe how data will be transformed and mapped to the target system from the source. A mapping rule describes how the source's data will be transformed and mapped to the target database's structure and format. Make sure that the two systems are on the same page when it comes to data formats, schemas, and data types.
5. Make a full backup of the source data, and plan for contingency scenarios and rollback procedures in case of unforeseen migration problems. You will learn how to do this in the next section of this lesson.
6. To validate mapping and transformation rules, perform a test migration with a subset of data. It is important to verify that the target system functions properly, and that the data is accurately transferred. When migrating data, **validation rules** are used to verify the integrity and quality of the data. By identifying and resolving data inconsistencies, missing values, and other issues that may affect the accuracy and reliability of the migrated data, you help ensure the data is accurate and reliable.
7. Perform the actual data migration once the testing has been successful. Maintaining data consistency during migration may require downtime or data freezing on the source system.
8. In the target system, verify that the data is complete and accurate after migration. By validating the data, you can ensure its accuracy and consistency. In order to improve the quality of your data, you can perform additional data profiling and cleansing as well as use checksums or hash checks on your data to detect any alterations. Backups and recovery procedures are essential to safeguarding data and quickly resolving integrity and completeness issues.
9. Perform post-migration testing to ensure that the target system is functioning correctly and meeting performance requirements. If any problems arise after migration, have a support team ready to assist.
10. The end users should be trained on the new system, and feedback should be gathered to address any usability concerns they may have.
11. Once the migration is complete, archive or purge the old data from the source system, freeing up space and reducing redundant data.
12. For future reference, it is important to document the entire migration process, including procedures, test results, and any lessons learned.

TERMS TO KNOW

**Data Migration**

The process of transferring data from one database to another while ensuring the data's integrity, accuracy, and consistency.

**Validation Rule**

A constraint that prevents invalid, inaccurate, or inconsistent data from being populated into database entities.

# 2. Backup to Avoid Data Loss and Corruption

A database migration involves inherent risks and complexities. Data migration involves moving large volumes of critical data between systems, which leaves room for data loss, corruption, and other problems. Backing up the original database creates a reliable and complete copy before migration begins. Backups ensure administrators can quickly restore the database to its original state in case of data loss or corruption during migration. This prevents disruption to business operations and financial or reputational damage.

HINT

Backups also offer a rollback capability, allowing organizations to revert to the pre-migration state if the migration encounters major problems or fails to meet expectations. With this flexibility, database

administrators can proceed with their migration plans without worrying about losing valuable data or compromising production environments' stability.

Furthermore, backups are essential to ensuring business continuity. During the migration process, there may be temporary downtime or a transitional database state. Backups enable organizations to quickly recover and restore services in case of unexpected delays or complications that prolong the migration timeline, ensuring uninterrupted access to crucial data and smooth operations. Backups generally provide a vital safety measure that mitigates risks and enhances confidence in the migration process, safeguarding valuable data and ensuring business continuity.

## 2a. Business Continuity

Business continuity refers to the ability of a business to maintain its normal operations and productivity levels regardless of what special situations are occurring, including not only planned events like data migration but also unexpected natural and human-created disasters. The following practices can help ensure business continuity throughout a database migration:

- Conduct a comprehensive risk assessment to identify potential challenges and issues during the migration.
- Engage key stakeholders, including business owners, IT teams, and end users, to understand their requirements and concerns.
- Plan out the scope, goals, timeline, and allocation of resources in a comprehensive migration plan.
- Ensure your plan addresses unforeseen circumstances by including contingency measures.

Migrating smoothly depends on a well-considered plan that minimizes disruptions and protects all aspects of the process.


You should create and verify a complete backup of the existing database before beginning the migration. The backup serves as a safety net in case of data loss or corruption during migration. Make sure backups are validated regularly to ensure data integrity and rapid recovery. You should also develop a rollback strategy for reverting to the original database state if the migration encounters major problems. In order to maintain business continuity and data availability, it is imperative to have a robust backup and rollback strategy.

Consider incremental or parallel migration approaches, where data is migrated in manageable batches. With this approach, business operations are less affected by migration, as it allows for continuous monitoring and validation. You might alternatively consider running the old and new database systems simultaneously during the transition phase with data synchronized between them. During the migration, users have access to up-to-date information because of the real-time synchronization of data between the two databases. Syncing and verifying all data is the key to minimizing downtime and ensuring a seamless switch to the new database.

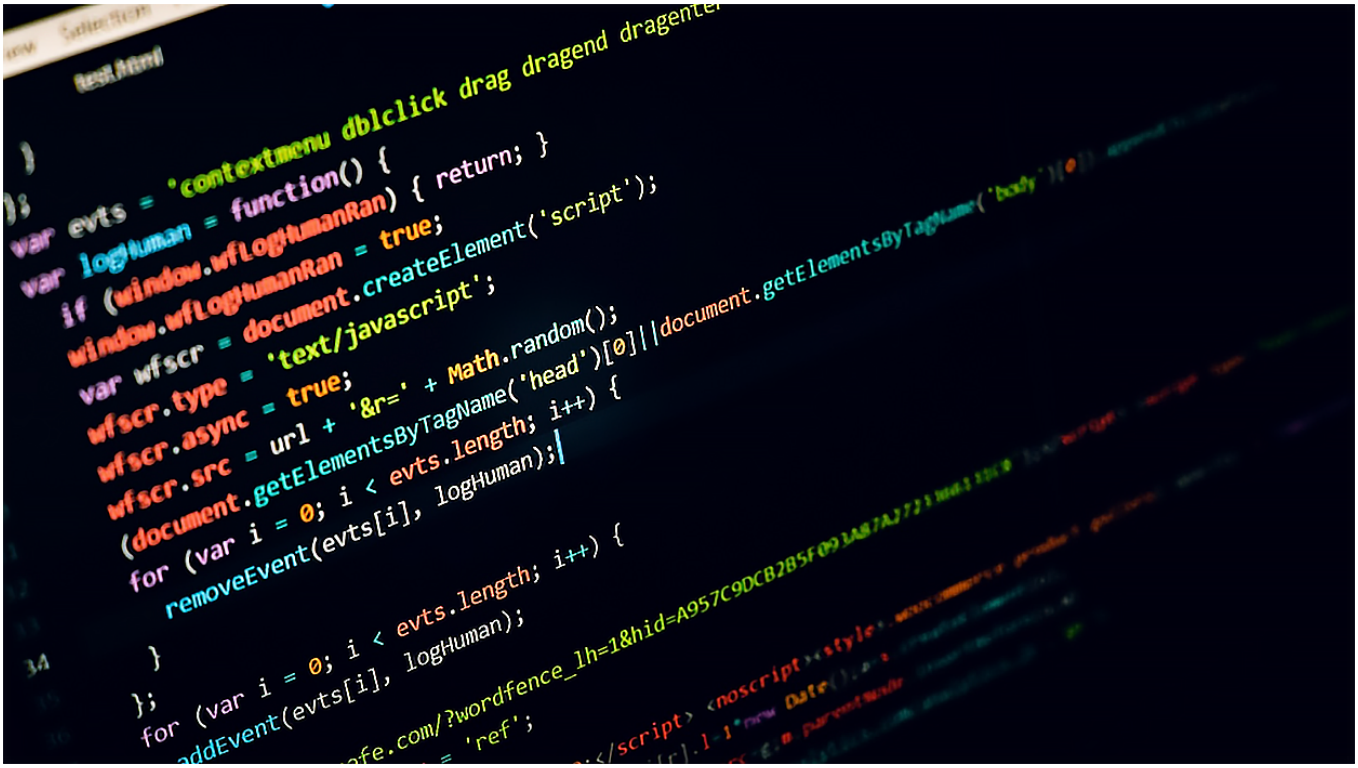
It is important for organizations to follow the steps outlined in this section to mitigate risks, maintain data integrity, and ensure the continuity of their business operations during database migrations. Planned testing, thorough backups, and robust rollback strategies facilitate a seamless transition to the new database environment, allowing businesses to continue operating uninterrupted.

## 2b. Data Security

Maintaining customer trust during a database migration is crucial for data security. This can be done by encrypting data, using secure servers, and limiting access to sensitive data. Additionally, organizations should keep customers informed of the progress of the migration and any steps they are taking to protect their data.

Protect data during migration by encrypting it to prevent unauthorized access or interception. Make sure the communication between the source and target databases is secure by using in-transit encryption protocols, such as TLS/SSL, that protect data while in transit so that it cannot be exposed or tampered with. You might also encrypt all data at rest on storage devices using folder-based encryption like NTFS's Encrypting File System (EFS) and/or whole disk encryption such as Windows' Bitlocker.

Ensure strict access control mechanisms are implemented to restrict the number of individuals with migration privileges. Follow the **principle of least privilege** when assigning roles and permissions to users involved in migration processes. To prevent unauthorized access to the database systems, use strong authentication methods such as **multifactor authentication (MFA)**. Detect and respond quickly to any suspicious behavior detected during the migration by auditing and monitoring user activity.




Protect sensitive or personally identifiable information (PII) before migration by using data masking or other anonymization techniques. With **data masking**, a realistic but fictional representation of the original data is used to replace sensitive data during migration, ensuring that the original data is not exposed. Masking data can prevent data breaches and comply with privacy regulations by enabling developers and administrators to work with realistic data.

Identify and address any security weaknesses by testing and assessing both the source and target databases before, during, and after the migration. Maintain a high level of data security throughout the migration process by engaging with cybersecurity experts and following industry best practices. Organizations can mitigate data

security risks and ensure a secure and successful database migration by implementing robust security measures and monitoring the migration closely.

## TERMS TO KNOW

**Principle of Least Privilege**

A security strategy that gives users only the minimum privileges they need to do their work.

**Multifactor Authentication (MFA)**

The practice of requiring two different forms of authentication for access to a system, such as a username/password and a PIN or access code.

**Data Masking**

A method of anonymization that replaces sensitive data with consistent, nonsensitive values.

## SUMMARY

In this lesson, you learned that the process of migrating data between two databases involves a series of steps in a **data migration plan** to ensure smooth and accurate data transfer. In most cases, the process begins with assessing the source and target databases to determine their structures, data types, and relationships. A plan is then developed, detailing the migration's scope, objectives, timeline, and resource requirements. Following the data profiling and cleansing process, the data will be transformed or corrected as necessary based on its quality and consistency. After that, the data mapping and transformation rules are defined, which describe how data is mapped from the source database to the target database. This step handles any differences between the two databases regarding data formats or schemas.

You also learned that it is important to validate the migration process and ensure data accuracy; a subset of data is migrated to ensure the mapping rules are in place. During this phase, any discrepancies or issues are addressed. Upon successful testing, the actual migration is carried out. A data validation and integrity check and a **backup to avoid data loss and corruption** is conducted before and after migration to ensure that the source and target databases are identical and quality standards are met. The new database is implemented after a successful migration, after user training has been provided, and after the documentation has been updated. The performance of the database may also need to be monitored and tuned periodically after the migration to ensure optimal performance. Data migration between databases can be efficient and secure if organizations follow these steps and best practices, minimizing disruptions and maintaining **business continuity**, **data security**, and data integrity at every stage.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND Faithe Wempen (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR TERMS OF USE.

## TERMS TO KNOW

**Data Masking**

A method of anonymization that replaces sensitive data with consistent, nonsensitive values.

**Data Migration**

The process of transferring data from one database to another while ensuring the data's integrity, accuracy, and consistency.

**Multifactor Authentication (MFA)**

The practice of requiring two different forms of authentication for access to a system, such as a username/password and a PIN or access code.

**Principle of Least Privilege**

A security strategy that gives users only the minimum privileges they need to do their work.

**Validation Rule**

A constraint that prevents invalid, inaccurate, or inconsistent data from being populated into database entities.