



Journal Homepage: - [www.journalijar.com](http://www.journalijar.com)  
**INTERNATIONAL JOURNAL OF  
 ADVANCED RESEARCH (IJAR)**

Article DOI: 10.21474/IJAR01/7050  
 DOI URL: <http://dx.doi.org/10.21474/IJAR01/7050>



### RESEARCH ARTICLE

#### DATA MIGRATION.

**Dhana Mulagala.**

Infrastructure Engineer, Washington DC.

#### Manuscript Info

##### Manuscript History

Received: 07 March 2018  
 Final Accepted: 09 April 2018  
 Published: May 2018

##### Keywords:-

Query processing, Database systems,  
 File systems, Data migration, Object  
 oriented databases, Data warehouse,  
 Analytics, Data integration.

#### Abstract

Modern computer systems are expected to be up continuously: even planned downtime to accomplish system reconfiguration is becoming unacceptable, so more and more changes are having to be made to “live” systems that are running production workloads. One of those changes is data migration: moving data from one storage device to another for load balancing, system expansion, failure recovery, or a myriad of other reasons. This document gives the overview of all the process involved in Data Migration. Data Migration is a multi-step process that begins with an analysis of the legacy data and culminates in the loading and reconciliation of data into the new applications.

*Copy Right, IJAR, 2018,. All rights reserved.*

#### Introduction:-

Migration is a process of moving data from one platform/format to another platform/format. It involves Migrating data from a legacy system to the new system, preferably without disrupting or disabling active applications and then redirecting all input/output activity to the new device. Data Migration is the also the process of importing data from various source systems into one target system. Data Migration is a multi-step process that begins with an analysis of the legacy data and culminates in the loading and reconciliation of data into the new applications. This process involves scrubbing the legacy data, mapping the data from the legacy system to the new system, designing the conversion programs, building and testing the conversion programs conducting the conversion, and reconciling the converted.

Why do we need to migrate data?

**In today’s world, migrations of data for business reasons are becoming common. While the replacement for the old legacy system is the common reason, some other factors also play a significant role in deciding to migrate the data into a new environment. Some of them are**

- **Databases continue to grow exponentially that requires additional storage capacity.**
- **Companies are switching to high-end servers.**
- **Need to reduce complexity and costs by moving to common / stable platforms.**
- **The concept of virtualization and the resultant need for data to be transportable, making it possible to migrate to and from physical and virtual environments.**
- **IT rationalization to facilitate compliance with the requirements of legislation**

**Data migration strategy should be designed in an effective way such that it will enable us to ensure that tomorrow’s purchasing decisions fully meet both present and future business and the business returns maximum return on investment.**

**Data Migration Strategy:-**

A well-defined data migration strategy should address the challenges of identifying source data, interacting with continuously changing targets, meeting data quality requirements, creating appropriate project methodologies, and developing general migration expertise.

The key considerations and inputs for defining data migration strategy are given below:

- Strategy to ensure the accuracy and completeness of the migrated data post migration
- Agile principles that let the logical group of data to be migrated iteratively
- Plans to address the source data quality challenges faced currently as well as data quality expectations of the target systems
- Design an integrated migration environment with proper checkpoints, controls and audits in place to allow fallout accounts/errors are identified/reported/resolved and fixed
- Solution to ensure appropriate reconciliation at different checkpoints ensuring the completeness of migration
- Solution to the selection of correct tools and technologies to cater for the complex nature of migration
- Should be able to handle large volume data during migration
- Migration development / testing activities should be separated from legacy and target applications

In brief, the Data Migration strategy will involve the following key steps during end-to-end data migration:

- Identify the Legacy / source data to be migrated
- Identification any specific Configuration data required from Legacy applications
- Classify the process of migration whether Manual or Automated
- Profile the legacy data in detail
- Identify data cleansing areas
- Map attributes between legacy and Target systems
- Identify and map data to be migrated to the historical data store solution (archive)
- Gather and prepare transformation rules
- Conduct Pre-Migration Cleansing where applicable
- Extract the data
- Transform the data along with limited cleansing or standardization
- Load the data
- Reconcile the data

1) *The Migration process steps will be achieved through the following SDLC phases*

*B. Requirement Phase*

Requirement phase is beginning a phase of the migration project; requirement phase states that the number and a brief description of systems from which data will be migrated what kind of data available in those systems, the overall quality of data, to which system the data should be migrated. Business requirements for the data help determine what data to migrate. These requirements can be derived from agreements, objectives, and scope of the migration. It contains the in the scope of the project along with stakeholders. The proper analysis gives a clear understanding of scope to begin, and will not be completed until tested against clearly identified project constraints. During this Phase, listed below activities is performed:

- Legacy system understanding to baseline the Migration scope
- Migration Acceptance Criteria and sign-off
- Legacy data extraction format discussion and finalization
- Understanding target data structure
- Legacy data analysis & Profiling to identify the data gaps and anomalies
- Firming up Migration Deliverables and Ownership

*C. Design Phase*

During the design phase, i.e., after Analysis phase, High-level Approach and Low-level design for Migration components are created. Following are the contents of the design phase.

- Approach for an end to end Migration
- Master and transactional data handling approach

- Approach for historical data handling
- Approach for data cleansing
- Data Reconciliation and Error handling approach
- Target Load and validation approach
- Cut-over approach
- Software and Hardware requirements for E2E Migration
- Data model changes between Legacy and target. For example – Change in Account structure, Change in Package and Component structure
- Data type changes between Legacy and target
- Missing/Un-available values for target mandatory columns
- Business Process specific changes or implementing some Business Rules
- Changes in List of Values between Legacy and target
- Target attribute that needs to be derived from more than one columns of Legacy
- Target Specific unique identifier requirement

The detailed design phase comprises of the following key activities:

- Data Profiling - Data profiling is an analysis of the candidate data sources for a data warehouse to clarify the structure, content, relationships and derivation rules of the data. The benefits of data profiling are to improve data quality, shorten the implementation cycle of major projects, and improve understanding of data for the users.
- Source to Target attribute mapping
- Staging area design
- Technical design
- Audit, Rejection and Error handling

#### *D. Development Phase*

After completing the Design phase, Migration team works on building Migration components as listed below:

- ETL mappings for Cleansing and Transformation
- Building Fallout Framework
- Building Reconciliation Scripts
- All these components are Unit Tested before Dry Runs.

Actual coding and unit testing will be done in the construction phase. During the development phase, the structures which are similar to target system should be created. Data from different legacy systems will be extracted into a staging area & source data will be consolidated in the staging area. The Construction and Unit Testing (CUT) phase include the development of mappings in ETL tool depend on Source to Target Mapping sheet. Transformation Rules will be applied in required mappings and loaded the data into target structures. Reconciliation programs will also be developed during this phase. Unit test cases (UTC) to be written for validating source systems, source definitions, target definitions and to check the connection strings, validation of data types and ports, etc.

#### *E. Testing Phase*

The objectives of the data migration testing are

- All required data elements have been fetched/moved.
- Data has been fetched/moved for specified time periods as per the requirement specification document.
- Data has originated from the correct source tables.
- The appropriate target tables have been populated in the correct format
- Performance of migration programs and custom scripts.
- There has been no data loss during data migration, and even if there is any loss, they are explained properly.
- Data integrity is maintained.

The Migrated data will be tested by reconciliation process. The reconciliation scripts will be developed to check the count between the source system and staging area where the data pulled from the source system, to check the count between source staging area, target staging area, and rejection tables. For example Count of source records = Count of target records + count of rejected records in rejection table.

All functional check validation will be verified depends on the business rules through functional scripts. The various ways to do the reconciliation and functional checks, it can be a manual process or automated process. Automation Process includes creating one procedure to check reconciliation which returns the records in error table if any of the scripts fail and creating one procedure to check the functionality which returns the records in error table if any of the function fails or creating macro to execute the reconciliation and functional scripts in one shot.

*F. Delivery*

The migrated data will be moved to QA environment to check the quality of data. During this phase, the cutover activities will be performed, and migrations from the legacy systems into Target Systems will be carried out. After the data is loaded, reconciliation of the loaded records will be verified. Reconciliation identifies the number of records that are successfully converted. It also tracks the records failed during the process of conversion and migration. The analysis will be performed on the failed records to determine the reasons for failure, and the necessary fixes are applied to the failed records to load them into the target database. The cutover phase resembles the final tasks in the SDLC implementation phase, including data conversion, testing, the changeover to the new system, and user training. The cutover phase is the final phase of the volume move.

During the cutover phase, the data in the source volume and the destination volume is synchronized. From the project perspective, the final preparation for cutover phase includes:

- Resolution of any critical issues (and documented in the upgrade script).
- Creation of the cutover plan, based on the results and experiences of the tests.
- An (optional) dress rehearsal test upgrade has been performed based on this cutover plan.

Phases in Data Migration

Phase 1 - Data Assessment		
Key Activities	Key Participating Groups	Deliverables/Outputs
<ul style="list-style-type: none"> <li>• Identification of data sources</li> <li>• Run system extracts and queries</li> <li>• Conduct user interviews and awareness programs on data migration process</li> <li>• Review migration scope and validation strategy</li> <li>• Create work plan and milestone dates</li> </ul>	<ul style="list-style-type: none"> <li>• Data migration leads</li> <li>• End / Business users</li> <li>• Program sponsors</li> </ul>	<ul style="list-style-type: none"> <li>• Scope document</li> <li>• Strategy document</li> <li>• Work breakdown structure with milestone dates</li> </ul>
Phase 2 - Data Cleansing		
Key Activities	Key Participating Groups	Deliverables/Outputs
<ul style="list-style-type: none"> <li>• Identify data cleansing needs and expectations</li> <li>• Create data prep worksheets</li> <li>• Clean up source data in current system</li> <li>• Format unstructured data in other systems</li> <li>• Run extracts and queries</li> </ul>	<ul style="list-style-type: none"> <li>• Data migration team</li> <li>• Client Information Support team</li> </ul>	<ul style="list-style-type: none"> <li>• Cleaned/changed source data that increases the success of automated data conversion</li> <li>• Control metrics and dashboards</li> </ul>

<ul style="list-style-type: none"> <li>to determine data quality</li> <li>• Create metrics to capture data volume, peak hours and off-peak hours</li> </ul>		
<b>Phase 3 - Test Extract and Load</b>		
<b>Key Activities</b>	<b>Key Participating Groups</b>	<b>Deliverables/Outputs</b>
<ul style="list-style-type: none"> <li>• Create/verify data element mappings</li> <li>• Run data extracts from current system(s)</li> <li>• Create tables, scripts, jobs to automate the extraction</li> <li>• Address additional data clean-up issues</li> <li>• Execute application specific customizations</li> <li>• Run mock migrations</li> <li>• Load extracts into the new system using ETL tools or SQL loader with bulk loading functions</li> <li>• Conduct internal data validation checks including business rules and referential integrity checks</li> <li>• Report exceptions to client team</li> <li>• Perform data validation</li> </ul>	<ul style="list-style-type: none"> <li>• Data migration team</li> <li>• Client Information Support team</li> <li>• DBA team</li> </ul>	<ul style="list-style-type: none"> <li>• Extracts from the source system</li> <li>• Data migration modules, jobs, scripts</li> <li>• Application loaded with converted data</li> <li>• Exceptions, alerts, and error handling control points</li> </ul>
<b>Phase 4 - Final Extract and Load</b>		
<b>Key Activities</b>	<b>Key Participating Groups</b>	<b>Deliverables/Outputs</b>
<ul style="list-style-type: none"> <li>• Run final extracts from the current system(s)</li> <li>• Execute specific customizations on the target database</li> <li>• Execute application specific customizations</li> <li>• Run pilot migrations</li> <li>• Load extracts into the new system using ETL tools or SQL loader with bulk loading functions</li> <li>• Conduct internal data validation checks including business rules and referential integrity checks</li> <li>• Report exceptions to client team</li> <li>• Perform data validation</li> </ul>	<ul style="list-style-type: none"> <li>• Data migration team</li> <li>• Client Information Support team</li> <li>• DBA team</li> </ul>	<ul style="list-style-type: none"> <li>• Data Extracts from the source system</li> <li>• Data migration modules, jobs, scripts</li> <li>• Exceptions, alerts, and error handling control points</li> </ul>
<b>Phase 5 - Migration Validation</b>		

Key Activities	Key Participating Groups	Deliverables/Outputs
<ul style="list-style-type: none"> <li>• Prepare migration validation reports and data movement metrics</li> <li>• Review migration validation reports and metrics</li> <li>• Record count of verifications on the new system</li> <li>• Reconcile any exceptions or unexpected variations on the data.</li> <li>• Sign off on validation</li> </ul>	<ul style="list-style-type: none"> <li>• Data migration team</li> <li>• Client Information Support team</li> <li>• Business users</li> </ul>	<ul style="list-style-type: none"> <li>• Signed-off migration validation document</li> </ul>
<b>Phase 6 - Post Migration Activities</b>		
Key Activities	Key Participating Groups	Deliverables/Outputs
<ul style="list-style-type: none"> <li>• Complete data migration reports and cross-reference files/manuals</li> <li>• Data correctness/quality reports</li> <li>• Target system reports and its correctness</li> <li>• Infrastructure capacity report and dashboards</li> <li>• Sign off</li> </ul>	<ul style="list-style-type: none"> <li>• Data migration team</li> <li>• Client Information Support team</li> <li>• Business users</li> <li>• Business sponsor</li> </ul>	<ul style="list-style-type: none"> <li>• Exception reports, cross-reference files/manuals</li> <li>• Infrastructure dashboards</li> <li>• Signed-off data migration project closure document</li> </ul>

**Test Approach for test runs**

Following are some of the best practices followed by test approach for mock runs / test runs to test the data migration activities before the crucial conversion to the production environment. Before entering the mock conversions, phase following criteria's must be met

- Tested the developed data conversion, cleansing and data validation programs
- Data Cleansed & transformed the legacy data for mock runs
- Developed & tested target system's load scripts/components
- Target system technical experts, subject matter expert's availability for Mock runs data loads into the target system & verification/validation
- Availability of the target system environment – like target production system
- Each mock run data load will be followed by validation and sign off by respective stakeholders for confirmation of validation and acceptance.

*G. Mock Run 1/Test run 1*

- Cleansed & converted representative legacy sample data (~40%) for a mock run
- Execution, reconciliation and load errors reports
- Conversion findings report & validation report for mock run 1
- Data & business SMEs verification & certification

*H. Mock Run 2/Test run 2*

- Fixes to the code / revise conversion approach from mock run1 observations
- Cleansed & converted representative legacy sample data loading (~80%) for a mock run
- Execution, conversion reconciliation and load error reports
- Conversion findings report & validation report for mock run 2
- SIT will be conducted as part of this test run. Data & business SMEs verification & certification

*I. MockRun 3/Test run 3*

- Fixes to the code / revise conversion approach from mock run 2 observations
- Cleansed & converted complete legacy data loading for mock run 3 (similar to production run)
- Execution, conversion reconciliation and load error reports
- Conversion findings report & validation report for mock run 3
- UAT will be conducted as part this test run, data & business SMEs verification & certification Following methods shall be utilized for validation of mock runs
- Visual check of data in the target system based on samples (comparison of legacy data and data loaded in the target system)
- Testing of an end to end processes with migrated data & system response
- Run target system specific standard reports/Queries for some of the transactional data
- Verification of load reconciliation reports and error report

**CutOver/Production migration approach:-**

Data Migration which is a subset of overall Migration can be achieved either using either Big-Bang approach or Phased Approach. Both the approaches have their own advantages and disadvantages. Comparison of Big-Bang vs. Phased Migration is listed below:

Big Bang Migration	Phased / Incremental Migration
No need to run two systems simultaneously reducing complexity and operation overheads	Multiple downtimes and high efforts
No synchronization issues. With a big bang migration, we don't need to keep the old environment up to speed with any record updates, the business has effectively moved on and is now being driven entirely from the target platform	Can work out for 24/7 operations scenarios that realistically cannot be shut down for several days/hours to migrate data, they need to stay online continuously.
Single Downtime and Low efforts	No worries of impact on Business during downtime and Cut-over Overrun
Fallback strategies can be challenging particularly if issues are found sometime after the migration event	It can be challenging to execute effectively due to the complexity of the underlying applications, data and business processes
Provides single downtime and low efforts but it adds Risk to the overall Migration due to the volume	Minimizes the Risk of Migration by moving a smaller chunk of Functionality & Data into Target platform but it adds complexity to overall Migration by having two Systems in parallel

Common Scenarios of testing

Below are the common scenarios which we can test during Data migration activities.

*J. Record count Check*

*K. Data in the staging will be populated from various source systems in general scenarios. One of the important scenarios to be tested after populating data in the staging is Record count check. Record count checks can be categorized into two sub-scenarios namely*

- Record count for Inserted Records: The count of record identified from source system using requirement document should match with the target system. i.e., Data populated after running the jobs
- Record count for updated records: The count of records which are updated in the target tables in the staging using data from source tables should match the record count identified from the source tables using requirement document.

#### L. Target tables and columns

If the jobs have to populate the data in a new table, check that required target table exists in the staging. Check that required columns are available in the target table according to the specification document.

#### M. Data check

Data checking can be further classified as follows.

- **Data validation for updated records:** With the help of requirement specification document we can identify the records from the source tables where data from these records will be updated in the target table in the staging. Now take one or more updated records from the target table, compare the data available in these records with the records identified from source tables. Check that, for updated records in the target table; transformation is happened according to the requirement specification using source table data.
- **Data validation for inserted records:** With the help of requirement specification document, we can identify the records from the source tables, which will be inserted in the target table available in the staging. Now take one or more inserted records from the target table, compare the data available in these records with the records identified from source tables. Check that, for inserted records in the target table transformation is happened according to the specification using source table data.
- **Duplicate records check:** If the specification document specifies that target table should not contain duplicate data, check that after running the required jobs, target table should not contain any duplicate records. To test this, you may need to prepare SQL query using specification document.
- **Checking the Data for specific columns:** If the specification specifies that particular column in the target table should have only specific values or if the column is a date field, then it should have the data for specific data ranges. Check that data available for those columns meeting the requirement.
- **Checking for Distinct values:** Check for distinct values available in the target table for any columns, if the specification document says that a column in the target table should have distinct columns. You can use a SQL query like “SELECT DISTINCT <COLUMN NAME> FROM <TABLE NAME>.”
- **Filter conditions:** When the job is completed, and data is populated in the target table, one of the important scenarios to be tested is identifying the criteria used to retrieve the data from the source table using specification/any other documents. Prepare SQL query using that criterion and execute on the source database. It should give the same data, which is populated in the target table. Filter conditions used to retrieve the data from source tables should match the criteria identified above.
- **Many source rows to one target rows:** Sometimes requirement specifies that, from many similar records identified from source, only one record has to be transferred to the target staging. For example, If the criteria used to retrieve the records from source tables gives 100 records, i.e., 10 distinct records are there, and each distinct record has 9 similar records, so total is  $10 \times 10 = 100$ . In this case, some requirement specification enforces that only 10 distinct records have to be transferred to the target table in the staging. This condition needs to be verified, and it can be tested by writing SQL statements. In some cases certain requirements specify that all the records identified from source have to be transferred to the target table instead of 1 distinct cord from several similar records, In this case, we have to check the data populated in the target table to verify the above requirements.
- **Check the data format:** Sometimes data populated in the staging should satisfy certain formats according to the requirement. For example, date column of the target table should store data in the format ‘YYYYMMDD.’ This kind of format transformation has to be tested as mentioned in the requirements.
- **Check for the deleted Records:** In Certain scenarios, records from the source systems, which were already populated in the staging, may be deleted according to the requirements. In this case, corresponding records from the staging should be marked as deleted as per the requirements.

#### File processing

In few scenarios, it is required that data in the staging has to be populated using flat file which is generated from a source system. Following tests have to be performed before and after executing jobs to process the flat file.

- Check for the directories where the flat file has to be placed for processing.
- Check for the directories where the flat file has to be moved after processing is completed.
- Check that file name format is as expected.
- Check for the record count in the target table. It should be as specified in the requirement.



- While populating data from flat file to the target table, Data transformation must be happened according to the requirements.

#### File Generation

Sometimes batch jobs will populate the data in a flat file instead of tables in the database. This flat file may be used as input file in other areas. There is a need to verify the correctness of the file generated by running the batch jobs. Following things need to verify as part of testing.

- Check that target directory is available to place the flat file generated. This check has to be performed before running the job.
- Since the generated flat file will be used as an input file to another system in the integrated environment, check that generated flat file format and file name format meets the requirement.
- Check that data populated in the flat file meets the requirement. Validate the data populated in this file against source data.
- Check that number of records populated in this file meets the specification.

#### Data Migration Risk:-

Risk management will be a part of every project, especially a large scale migration project. Risks related to schedule, costs, and scope should be clearly identified, documented and a plan must be put together to mitigate them. Risks are specific to data migration come from a number of contributors like

#### *N. Data quality*

*O. There is high risk associated with the understanding of business rules & logic related to flow of information between the source and target systems. Data from the source system may not map directly to Target system because of its structure, and multiple source databases may have different data models. Also, data in source databases may not have consistent formatting or may not be represented the same way as the Target system. There will be lack of expertise in source data formats. Understanding how data has been formatted in the source system is critical for planning how data entities will map to Target.*

#### *P. Extraction, transformation, and load tool complexity*

Business systems will have more functionality and modules, and this becomes an extremely challenging exercise. In earlier days, data migrations of metadata were easily accomplished by writing data directly into the database(s) by using simple commands and flat files at the database layer. As business systems have matured, most of the times the database layer doesn't contain the hierarchy and relationship information, these are stored and managed within the application layer. ETL have to use APIs, and in some cases, these are not recommended to support extraction and loads in a mass manner

#### *Q. Performance of systems (extraction and target system persistence)*

The poor performance of application could lead delay in a load of data, hence through testing with multiple runs and cycles are essential to figuring out the exact performance of the system. Load test and stress tests need to be carried out to check the feasibility of the application to run on a high volume of data during peak hours. Hence IT managers should ensure that the best performance people are monitoring the load process and can tune the systems properly. From a database perspective, there might need to increase the memory allocated or turning off archiving or altering the indexing

#### *R. Project coordination*

Most data migration projects will be strategic for the organization and will be used to implement the new business system. This involves multiple roles to be played during analysis, design, testing, and implementation. It requires dedication and well-planned coordination to meet the schedule of data migration as any slippages in any of the phase will impact the overall project.

#### Data Migration Challenges and Possible mitigations:-

Data migration activities are no longer a threat to IT organization. IT manager's uses best practices followed in the industry, technology-driven focus, and domain experience to tackle the task of data migration. The process of migrating data can be divided into the smaller task by which the process and procedures are

controlled and will help the organization in reducing the cost and time to completion. Below are the few challenges that might pose a threat to the data migration activities.

S.No	Potential Challenges	Mitigation areas / accelerators
1	Legacy data fitment and hidden data quality issues	Conduct data quality assessment to understand issues of migration. Leverage any data quality assessment framework
2	Data staging and integration from various sources. Source data is not available in specific formats	Define source-neutral data formats/models to hold legacy data
3	Business rules for data conversion cleansing	Legacy system SMEs availability to review the data quality issues and define business rules at the initial stages of the project
4	Data Re-conciliation	Re-usable re-conciliation and audit framework & design
5	Data validation and manual data enrichment by the data stewards/owners	Information platform web application to view/verify exceptions and take appropriate decisions on the error records by data stewards/owners/subject matter experts
6	Unexpected load failures of the cleansed & converted data at the time of target load during mock runs. Also, leading to increasing the number of mock runs / increase in the mock run cycle period	Build pre-target load validation rules within the staging area to identify target load issues in advance
7	Incomplete/Improper data in the target system post loading	The process to automate the post data load verification
8	Unavailability of load error analysis reports	The process to automate and generate error log analysis reports

## Performances and Best Practices:-

### S. Performance

Performance of the migration can be optimized in both staging area and ETL jobs. Some of the best practices followed in the industry are given below.

#### 1) General

- **Standard best practices followed by ETL tools for better performance should be adopted for jobs development. Each ETL tool has few tips to enhance the performance which can be reused with an understanding of the tool's capabilities can aid in this. Parallel processing and pipelining and allow data to be partitioned are few tips that can improve the speed or performance.**
- **It is important to keep in mind that ETL tools have been designed to deal with large volumes of data. A join in the database query across multiple tables with large volumes embedded inside a source/SQL stage in the tool can prove to be 'expensive' and may take time to return results. Instead, use the stages provided by the ETL tool to split this query and fully utilize the application database as well. In this case, you can use a join stage so that the database merely fetches data from each table leaving the heavy processing to the ETL tool.**

#### 2) Extract Phase

- **Breaking down the jobs into small technical components to enable a maximum number of jobs in parallel.**
- **Creation of a special table on the legacy system to maintain the list of customers, contracts, accounts, connection objects and meters that have to be migrated. This would have enhanced performance.**

#### 3) File Generation Stage

- Creation of indexes to optimize the creation of these flat files.

T.

### U. Best Practices

Some best practices for an ETL tool and staging database project are as follows:

#### 1) ETL Tool

- Parallel Threads in Query
- The degree of Parallelism of flow of data
- **Parameterization:** Use the Substitution Parameters instead of Global Variables for constants whose values change only with the environment.
- **Temporary tables:** Temporary tables which are created at runtime and then act as normal tables will help in converting complex business logic into multiple data flows and populate template tables, which can be used for lookups and/or joins.
- Monitor Log
- Collect Statistics

#### 2) Database

- Indexes
- Table Space: Use different tablespaces for creating tables as per data volumes, which helps in distributed data storage
- Avoid using 'NOT IN.'
- A number of loaders: The number of loaders can be increased while loading data during jobs as it improves loading performance.
- Avoid use of Select \*
- Database utilities: Use the built-in utilities to extract table definitions from the staging database.

Hidden Cost involved in Data Migration activity

While some of the costs associated with data migration might be apparent, such as the purchase price of the necessary migration tools, others are not so obvious. There should be a proper plan for effective migration as all the stakeholders should understand and appreciate the hidden costs involved during migration.

*V. Planned & Unplanned Downtime*

Scheduled & Unscheduled downtime is expensive due to non-availability of applications or data. This will eventually hit the profits of the organizations. We need to make sure that both the application and data are available throughout the migration to avoid any costs associated due to downtime. As a mitigation plan, IT managers can plan to do migration on minimal disruptive operations.

*W. Data Quality*

The need for data migration is to have good data quality for the application. If the quality of data in target database / data warehouse is not up to the expectations or standards the purpose of the entire application is not satisfied. Bad data quality can cripple an application long before roll-out. If the data fails to pass a base set of validation rules defined in the target application, the data load will fail which will increase the cost of rework and takes more time for go-live activities. Organizations must establish user confidence in the data. In order to fully trust data, business should have traceability of each element of data through data profiling, validation, and cleansing processes.

*X. Human hours*

Staff time is more precious, and it is safer to schedule the migration activities during holidays or weekends to have minimal impact on the businesses. This will certainly reduce the downtime needed during critical hours. To reduce the overtime cost, business should look ways to zero downtime migration. There is no guarantee that all the steps will reduce the necessity for overtime, but certainly, it will also better place the IT department to deal with certain problems without the need for disruptive and expensive unscheduled downtime.

*Y. Loss of data*

One of the common problems faced in data migration process is the data loss and the frequency of the data loss. On an average, about a third of businesses have lost data during migration with about a half of those data losses directly translating into monetary losses. Migration strategies should be formulated to mitigate any risk associated with loss of data. Best mitigation policy is to use full data backup of the source system before migrating to the target system.

*Z. Failure to Validate*

A large number of businesses fails to validate the results of their migrations properly. They solely trust the results of the users testing and not the data that actually got moved out through migration. This will have ripple effect in terms of delay in the identification of problems and these results in either expensive unscheduled downtime during business hours or further evening or weekend overtime. To avoid such situations, businesses should seek to deploy a solution has inbuilt validation capabilities.

*AA. Under budgeting*

Proper planning on man hours, downtime, etc. are required for arriving at the cost of data migration to avoid any surprises on the cost involved in the data migration Sometimes it will be either over budgeted or under budgeted and according to industry analysts in two thirds of the migrations companies calculate wrong man hours/downtime requirements.

**Roll Back Strategy:-**

Rollback strategy is mandatory for any data migration process as a mitigation plan to restate the application activities in case of any unforeseen situations or failure during migration activities. This has to be planned between initial pilot phase and throughout the entire phases of actual migration. This has to be planned for every level specified in the migration planning. This strategy will help in describing all the activities to be executed If a process fails or does not produce an acceptable result.

On a broader note, we should plan for the following activities

- Plan to clean the target system data when the target is also in a production environment.
- How to push the transactions during the migration / cut-over window into legacy applications
- Plan to minimize the impact on downstream / upstream applications.
- Restore the operations from existing or legacy systems
- A communication plan for all stakeholders

#### **Reconciliation:-**

**Data audit and reconciliation is a critical process in determining the overall success of a data migration event. Reconciliation is the process of keeping migration activities such as Cleansing, Extraction, and Loading under control. It is also instrumental in keeping migration on track, ensuring the authenticity of the quality and quantity of the data migrated at each stage. Validation programs have to be introduced at the extract and upload levels to capture the count of records. Analysis and Reconciliation (A&R) checks for accuracy and completeness of the migrated data and deals with quantity / value criteria (number of objects migrated etc.). A&R does not include verification of the migrated data to make sure that the data supports the business processes. Reconciliation requirements are driven by both the SAP functional teams -Process Owners requirements and the Technical reconciliation requirements. Below are the tasks involved in A&R process.**

- **Run reports at source:** A&R reports can be executed at legacy and number of records extracted will be validated against the legacy system. Hash counts for key columns are noted.
- **Extract Validation:** Data Extracts can then be validated against the Data Acceptance Criteria. This will also include validation of a number of records and hash count mentioned in Source A&R reports
- **Run intermediate reconciliation in transformation layer:** For some objects, intermediate reconciliation can be performed at the data transformation layer.
- **Run reports at target:** Target A&R reports must then be executed to validate the record count and hash counts

#### **Conclusion:-**

**Data Migration is the process of moving data between different storage types, formats, or computer systems. It is often required when organizations change computer systems or upgrade to new systems. This solution is usually performed through programs to achieve an automated migration. As a result, legacy data stored on out-of-date or obsolete formats are evaluated and then migrated to newer or more cost-effective & reliable storage Area. Data will be migrated into target what target system wants, depends on the need of target, data will be transformed from legacy systems and will be loaded into the target. Data migration is a regular activity of IT department in many organizations. It often causes major issues due to various reasons like the staff, downtime of the environment, the poor performance of the application and these will, in turn, affect the budgets. To prevent these types of issues, organizations need a reliable and consistent methodology that enables them to plan, design, migrate and validate the migration. Further, they need to evaluate the need for any migration software/tool that will support their specific migration requirements, including operating systems, storage platforms, and performance. In addition, migration products that maintain continuous data availability during the migration without affecting performance are desirable. To keep in check all the points, an organization needs robust Planning, Designing, Assessment and proper execution of the Project and its variables.**

**References:-**

1. E. Anderson, J. Hall, J. Hartline, M. Hobbs, A. Karlin, J. Saia, R. Swaminathan, and J. Wilkes. "Experimental Study of Data Migration Algorithms," 5th Workshop on Experimental Algorithms, August 2001.
2. ANSI, "Fibre Channel Arbitrated Loop," Standard X3.272-1996, April 1996.
3. T. F. Abdelzaher, "An Automated Profiling Subsystem for QoS-Aware Services," IEEE Real-Time Technology and Applications Symposium, pp. 208-217, June 2000.
4. T. F. Abdelzaher and N. Bhatti, "Web Server QoS Management by Adaptive Content Delivery," International Workshop on Quality of Service, pp. 216-225, 1999.
5. J. R. Douceur and W. J. Bolosky. "Progress-based regulation of low-importance processes," 17th ACM Symposium on Operating Systems Principles, pp. 247-260. Dec 1999.
6. G. F. Franklin, J. D. Powell and M. L. Workman, Digital Control of Dynamic Systems (3rd Ed.), AddisonWesley, 1998.
7. R. Golding, P. Bosch, C. Staelin, T. Sullivan, and John Wilkes. "Idleness is not Sloth," Winter'95 USENIX Conference, pp 201-212, Jan. 1995.
8. Y. Lu, A. Saxena, and T. F. Abdelzaher, "Differentiated Caching Services; A Control-Theoretical Approach," International Conference on Distributed Computing Systems, pp. 615-622, April 2001.
9. T. Madell, Disk and File Management Tasks in HP-UX, Prentice-Hall, 1997.
10. J. Nagle. "On packet switches with infinite storage," IEEE Trans. on Communications, vol. 35, no. 4, pp. 435-38, April 1987.
11. G. Papadopoulos, "Moore's Law Ain't Good Enough," keynote speech at Hot Chips X, August 1998.
12. S. Parekh, N. Gandhi, J. L. Hellerstein, D. Tilbury, T. S. Jayram, J. Bigus, "Using Control Theory to Achieve Service Level Objectives in Performance Management," IFIP/IEEE International Symposium on Integrated Network Management, 2001.
13. Thota, S. (2017), Big Data Quality. Encyclopedia of Big Data, pp.1-5. [https://link.springer.com/referenceworkentry/10.1007/978-3-319-32001-4\\_240-1](https://link.springer.com/referenceworkentry/10.1007/978-3-319-32001-4_240-1).
14. D. Steere, A. Goel, J. Gruenberg, D. McNamee, C. Pu, J. Walpole, "A Feedback-driven Proportion Allocator for Real-Rate Scheduling," Symposium on Operating Systems Design and Implementation, pp. 145-158, Feb 1999.