

**ERP Database Migration From On-Promise To The Cloud: An Iterative Methodology**

**Majda EL MARIOULI, Jalal LAASSIRI**

**Majda EL MARIOULI, Jalal LAASSIRI- “ERP Database Migration from On-Promise To The Cloud: An Iterative Methodology.” – Palarch’s Journal of Archaeology of Egypt/Egyptology 18(7). ISSN 1567-214x**

***Keywords:* Enterprise Resource Planning (ERP), ERP Cloud, Database migration, Data migration, ETL, Agile methodology****.**

**Abstract**

One of the most complex and demanding processes of migration from one or multiple legacy systems to an ERP system is database migration. Data migration is seen as a major research challenge in cloud environments. There are many factors to take in consideration on the migration process, there factors include: Data migration costs, data quality, duration, data volume, system performance, data retention and the enablement of new system functionality are all factors to take in consideration on the migration process. There are multiple cases to migrate data into an ERP Cloud, such as migrating and conversion from on-premise ERP to Cloud ERP for homogeneous or heterogeneous databases or from non-ERP systems to a Cloud ERP systems. Through this article we present different methodologies and approaches for data migration from legacy systems to modern systems. In addition we introduce an iterative methodology to migrate database from legacy ERP systems to ERP cloud using ETL

1. **INTRODUCTION**

 Modernizing legacy systems is one of the most challenging processes faced by many organizations. There is a huge need to migrate local systems to new systems at some point, especially with the emergence of new technologies and the evolution of application areas [1]. Migration to the cloud can be the subject of several factors, namely: the expansion of the global market, the need for a standard architecture, software license renewals, expiration of the data center lease, upgrades hardware level required, localization requirements to meet regulatory compliance, increased developer productivity.

 Cloud ERP offers a revolutionary solution to provide a flexible, adaptable, scalable, efficient and manageable ERP system. Cloud ERP as a business management software has provided big success to deliver business critical data [2]. It seems to become a real substitute to on-Premise ERP and firms would be likely “pushed” to switch toward the cloud solution [3].

Database migration, as a mainstay of legacy systems modernization projects, has been recognized as inherently complex. It is of critical importance to design internal controls that provide accuracy of the data selected for migration. To succeed in a migration project, a company needs to study and understand the different migration approaches and choose the approach that is most suitable for its case.

Database migration, as a fundamental aspect of projects on modernizing legacy systems, has been recognized to be a complex in their nature. It is of critical importance to design internal controls that provide accuracy of the data selected for migration.

 In some cases, transformations on data can lead to transformations on views, features, and user interfaces. However, we limit the scope of this article to transformations on data models and their corresponding schema.

 This paper is organized into five sections: the introduction as the Section One. In section two we provide a background of related works from previous research and E-books of Cloud ERP solution providers. Third section gives general information about database migration methodology and approaches. In the fourth section, we present an application of migrating two websites from two ERP databases to the single database of a modern ERP Cloud using an iterative methodology. The final section concerns the conclusion of this research.

1. **BACKGROUND**

P. Pant, S. Thakur et al, in their research which explores the issues and methods of data migration across the Clouds [4], divided database migration into the following categories:

- Database schema migration,

- Data migration,

- Database stored program migration,

- Application migration,

- Database administration script migration.

 The same categories are discussed by Laszewski and Nauduri in their book [5] about migrating to the Cloud: Oracle Client/Server Modernization. The book highlights a range of topics, from Oracle application and database cloud offerings, to the migration of any infrastructure to the cloud. Various migration tasks such as schema, data, stored procedures, and application migration are covered individually in many chapters.

 At the start of data migration activities, it is important to fully understand the volume of data in existing systems and other relevant technical and non-technical information. Following, 14 key questions with the biggest impact on the database migration effort that can help in migration task listed by Laszewski and Nauduri:

1. OS type and version, Database version and hardware vendor and specification
2. Online transaction processing (OLTP) or data warehousing or BigData
3. Number of users,

peak transaction

rate per second,

and daily data volumes

1. Database size
2. Number of stored procedures and total lines of code Stored
3. Number of triggers and total lines of code
4. Number of views and total lines of code
5. Application languages and database connectivity layer
6. Number of SQL statements in the application
7. DBA scripts
8. Replication server or products used
9. ETL product usage.
10. Backup and recovery and disaster recovery solutions
11. Sybase Open Server, Microsoft SSIS, IBM MQ Series, or other vendor- specific products

 B. Thalheim and Q. Wang [1] investigate data migration fundamentals from a theoretical perspective. Their study provides answers about data migration in practice such as

controlling data quality, reacting to specification changes and comparing legacy data sources with the migrated data in new systems.

 S. Vida, I. Pihir, R. Fabac listed the steps of the migration process from one ERP system to another in real time [6]. The fundamental starting point of the method described in his article is the definition of a set of X data, the migration of which from the current base to the base of the new ERP system makes it possible to start and continue the operation in the new system. The experience of authors of this paper [6] in the practical implementation of the described methods prove the fact that X-set does not necessarily encompass all data from the current database and the actual business year, but rather the data that is capable of determining the relevance of the data that has migrated to the new ERP system. As well as monitoring company operations in the new system at a given time of x-date + 1 during production.

 In the case of moving data between heterogeneous databases, G. wang and Z. Jia [7] has developed a general data migration model that can be used to set up an efficient data migration process.

 In the following section, we focus on investigating available methodologies and techniques in databases migration from legacy systems to ERP Cloud

1. **Methodologies of migrating databases to the Cloud**

 The migration from one database platform to another requires applications refactoring depending on the technical differences between the different databases.

 According to the migration project discussed in [Laszewski](http://b-ok.xyz/g/Tom%20Laszewski) et al. Book, “Fig. 1” represents detailed phases in the life cycle of the migration process.

 Based on [Laszewski](http://b-ok.xyz/g/Tom%20Laszewski) and [Nauduri](http://b-ok.xyz/g/Prakash%20Nauduri)’s methodology shown in “Fig. 2”, Strauch, Andrikopoulos and Karastoyanova et al [9] propose a vendor-independent and database technology-independent step-by-step methodology. This last one refines and adapts the one proposed in [5]. The steps of the methodology described in the figure below:





Fig. 1. Life cycle of a migration project detailed phases



Fig. 2. Methodology to migrate the database layer to the cloud

Next, they study the available vendor-specific, vendor-independent methodologies and guidelines for database layer migration:

* Amazon proposed a phase-driven approach to migrate an application to its cloud infrastructure comprising the following six phases: cloud assessment, proof of concept, data migration, application migration, leverage the cloud, and optimisation.
* A Windows Azure SQL Database Migration Wizard and the synchronisation service Windows Azure SQL Data Sync.
* The tool Bulk Loader offered by Google App Engine which supports the import of CSV and XML files into the App Engine Data Store and the export as CSV, XML, or text files.

The migration of a large database of ERP systems involves several steps that can be repeated iteratively. Some steps can be repeated - steps such as solution design architecture, data target modeling, data movement and transformation mapping, development and testing. Data profiling and requirements collection follows the iterative cycle, while deployment does not, see “Fig. 3”.



Fig. 3. Cyclical process for data migration solution.

According to [8] there are two approaches to ERP database migration. Both approaches have their own specific complexity challenges. These approaches are as follows:

* The transaction-driven approach is considered the most common approach for migrating an ERP migration. The data is taken from the source system and converted into a load file which in turn can be loaded using standard load programs. Compared to the table-based approach, the scope of data migration is limited to live data and is therefore more cost effective.
* The table-based approach involves the complete transfer of tables without any selection criteria. The table-based approach is successfully implemented in an ERP environment based on their experience for a multinational client.

These two approaches are compared based on those pillars: Accuracy, Completeness, Compliance, and Efficiency. The result of the comparison is shown in the table below in “Fig.4”.

Note that these methodologies are all commonly performed with data integration technologies such as ETL (extract, transform, and load), replication, and manual coding. Some projects require more than one of these data movement technologies and techniques. However, regardless of the type of data movement technique selected for a project, data migration is necessary:

* Use ETL-based tools for most data migration solutions; it is the most preferred technology for data migration.
* Profile source data carefully.
* Expect data migration
* Use an iterative process.
* Allocate significant time to modeling the target and mapping data to it.
* Do not forget about stored procedures and other procedural logic in the database.
1. **ERP database migration using an iterative methodology**

 In this section we provide a detailed description of the methodology used to migrate multiple websites running in multiple ERP systems which means multiple databases to a modern ERP system website with a single database.

 Due to the high maintenance, operation and performance costs of running multiple legacy ERP systems in different server instances and their rigidity to support business initiatives. The customer wanted to migrate the data from these systems to a common modern version of the same ERP under the PostgreSQL database.

 The main objective of this method is to migrate data from legacy systems to modern systems; Data profiling and analysis of existing data; Improve user operations and customer support; Reduce operating and maintenance costs; offering high scalability and performance

effective solution.

 The scope of the project was to migrate two websites developed in Odoo ERP system version 11 to an upgraded version of the same ERP. The targeted system is Odoo Version 13, which provides the possibility to run multiple websites in a single database. Fig. 5 represents the scope of this project.

 The client uses the native community version of Odoo 11 without any customisation. Many modules are installed and used for the best presentation of their websites such as: shop, events, slides, blog, contact, SEO, The first website is in French and English language and the second website is in Japanese and English Language.

 We used an agile development lifecycle methodology in the migration development process. Operation built in iterations that is reviewed by the owner of the final business.



Fig. 4. Comparison between Transaction driven approach and Table driven approach



Fig.5. the scope of the migration project

 To prepare the upgrade to the newly version of Odoo, we started by following steps:

1. Allocate a new server instance for test environment,
2. Install compatible version of PostgreSQL with Odoo 13,
3. Install Odoo 13 and it’s requirements,
4. Create a new database,
5. Install all modules similar to installed modules into legacy systems,
6. Configure the multi-website and multi-company in the new system and create the two websites with multi-languages, domains and themes..
7. Planning for the migration.

The stage of planning the migration contain following phases:

1. Gather requirements by identifying the different data sources of the existing targeted database. Also by analyzing the source data.
2. Landscape analysis to get an overview of the source and target systems in which the operation of each system has been understood, and how data is structured / modeled and the interaction between different models.
3. Data Assurance to review / validate existing data in our source systems if this has been done, to determine if it is suitable to transform them into target systems using landscape analysis. This step consists of two sub-steps:
* Data Profiling and Mapping: This is a document created to map the fields of the source and target systems, prepared by determining whether the historical data present in the source systems is suitable for migration to the new systems and also removing unnecessary historical data.
* Data Cleansing: Data quality checking in source systems as if data is available, complete and in correct format has been performed. If there is redundancy in the data migrated from both sources to the source database, it should be removed.
1. Core Migration: this phase can be divided into several mini steps:
* Migration Design: According to the mapping definitions presented in the mapping document, many fields are added in the new version of the system and others are removed, with the need to know what data should we put in the new fields.
* Testing Design: Integration testing and functional testing to verify whether the migration design is aligning with the business rules. In our case many adjustments should be made in the websites view pages because the version of Bootstrap used in the new system is changed from the Bootstrap used in legacy systems.
* Migration Development: This step concerns the use of Talend tool to extract, transform and load data into the target system.
	+ Extract: Extract data from the existing data sources according to the mapping document created from the above steps and made available for further processing.
	+ Transform: Filtering data by selecting the desired columns, calculating new fields based on the extracted data, joining data from different tables then performing aggregation operations.
	+ Load: Load the transformed data into our target database system: i.e. PostgreSQL database
* Testing Development (User Acceptance Testing): Check the number of records retrieved by a particular script, check whether the extracted data is formatted correctly in the target system, check whether the source data fields are accurately mapped to the fields in the target database, check whether the integrity and data quality is maintained. If all tests are correct, the new system has been deployed in a pre-production environment for business owners.
* Business Sign Off: Wait for the business owner to provide sign-off.
* Go Live and Production Support: After the successful completion of the basic migration test phase and various sprints, the new system was released for production and further enhancements were made to the system based on changes in business requirements.

The agile development life cycle has been followed and all corrupt migrations were caught at early stages of development which were rectified in the parallel sprints by changing the project time-line. Below, a resume of the iterative methodology steps “Fig. 6”.

The proposed iterative methodology was successful to migrate website modules and related modules from two legacy systems to a new version of the same system. The same steps can be followed to migrate other modules installed in the client legacy systems.

Fig.6. Iterative methodology for data migration using ELT Solution

1. **RESULT AND DISCUSSION**

 This chapter focuses on the migration results eventually obtained, the conclusion from these results and lastly the recommendations that are suggested.

**Results**

 The iterative development lifecycle methodology was employed for data migration which helped in achieving project objectives on time and in budget. Table I below provides an overview of the duration for each of the steps for the migration of two databases. These databases concern two websites running on an on-premise ERP system which have been migrated to one website in a single database using the proposed iterative methodology.

 According to the duration table, we can see that several steps were not necessary during the migration of the second website and other steps are done in a minimum duration. This is signified by the fact that processing is just repeated in the migration of the second website and the mapping between the columns of the tables was already done. The migration of the two websites from on premise ERP to a Cloud ERP took around a week with the iterative methodology. The same steps can be repeated for the migration of other modules, if we take into consideration only the native version without customizing the code.

 The results show that the iterative method allows the company to reach production in a minimum of time. It also provides the company with a version of the system to benefit from the Cloud and the new system version’s functionalities.

Data migration is not a process that can be executed one-time. Neither is it an open and shut case. Future changes are inevitable and Vendors must update technology by applying best practice and proven methodologies to deliver high quality, cost-effective solutions to Chief technology Officers.

Table- I: Overview of duration of the iterative methodology when migrating database

|  |
| --- |
|  |
| ***Iterative methodology Steps*** | ***Duration in hours for website 1*** | ***Duration in hours for website 2*** |
| Project preparation | 2 | 1 |
| Requirement analysis | 1 | - |
| Data modeling | 3 | 1 |
| Data cleansing | 3 | 3 |
| Migrating design | 6 | 6 |
| Testing design | 2 | 2 |
| Migrating development | 6 | 2 |
| Testing development | 2 | 2 |
| Business sign off | 1 | - |
| Go live and production support | 2 | 1 |

1. **CONCLUSION**

 Moving database workloads to the cloud is a worthwhile effort as part of modernizing the system architecture, it takes time and money to get data to the cloud server. The decision to move from legacy systems to ERP Cloud servers will help companies to improve their performance and business. Because they will have access to the best software and hardware in database management.

 We presented some general methodologies and approaches to migrate databases, then we introduced an iterative methodology used to migrate two databases of legacy systems to a Cloud ERP with a single database. We plan in the future to build a method which can be extended to support system evolution and system migration in a broader sense.

1. **REFERENCES**
2. B. Thalheim, Q. Wang. “Data Migration : A Theoretical Perspective”.[Data & Knowledge Engineering](https://www.sciencedirect.com/science/journal/0169023X), [Volume 87](https://www.sciencedirect.com/science/journal/0169023X/87/supp/C), September 2013, Pages 260-278.
3. G. Raihana “Cloud erp – a solution model”, International Journal of Computer Science and Information Technology & Security, 2012.
4. K. Mezghani, “From On-Premise ERP to Cloud ERP”, Encyclopedia of Information Science and Technology, Fourth Edition, 2017, Pages 2965-2974.
5. P. Pant, S. Thakur, “Data Migration Across The Clouds”, International Journal of Soft Computing and Engineering (IJSCE), 2013.
6. Laszewski, T. and Nauduri, P. (2011) Migrating to the Cloud: Oracle Client/Server Modernization, Elsevier, USA.
7. S. Vida, I. Pihir, R. Fabac, “Method of data migration from one ERP system to another in real time”, 21st Central European Conference on Information and Intelligent Systems, 2010
8. G. Wang, Z. Jia, “Data Migration Model and Algorithm between Heterogeneous Databases based on Web Service”, Journal of NETWORKS, VOL. 9, NO. 11, NOVEMBER 2014
9. R. Akça, MSc, https://www.compact.nl/articles/erp-data-migration/.
10. *S. Strauch, V. Andrikopoulos, D. Karastoyanova, “Migrating Enterprise Applications to the Cloud : Methodology a Evaluation Migrating enterprise applications to the cloud : methodology and evaluation”. Int. J. Big Data Intelligence, Vol. 1, No. 3, 2014*

**AUTHORS PROFILE**



 **Majda EL MARIOULI** Member of Computer Science Research Laboratory Ibn Tofail University Kenitra, Morocco. Phd student Working on the implementation of ERP systems into Cloud infrastructures And Technico-functional consultant specialized in ERP systems Her research interests include ERP systems implementation methodologies ERP migration from on-premise to Cloud infrastructures Refactoring ERP when migrating to the Cloud, ERP systems Security, ERP systems and artificial intelligence, ERP systems and machine learning.

D**r. Jalal LAASSIRI** Member ofComputer Science Research Laboratory at Ibn Tofail University, Kénitra, Morocco. Laboratory Team Head. Received his Ph.D. degree in computer sciences and engineering from University of Mohammed V, Rabat, Morocco. Member of the International Association of Engineers (IAENG), Dr. Laassiri authored/co-authored many Journal articles, Conference Proceedings and Book Chapters. His research interests include Network Security, Wireless Sensor Networks, Business Intelligent, Internet of Things, Big Data, Smart-Cities; Professor since October 2010 at the Faculty of Sciences of Kenitra, Department of Computer Science, Ibn Tofail University, Morocco.