Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 1, Jan – Mar 2026 || PP. 50 – 57 https://sjaibt.org

AI-Assisted Schema Transformation for Automated Legacy-to-Cloud Database Migration

DOI: https://doi.org/10.63345/sjaibt.v3.i1.301

Sarvesh Kumar Gupta
Consulting Member of Technical Staff
Oracle
Saint Peters, Missouri -63376 , USA
ORCID: 0009-0008-7460-4874

Date of Submission: 07-02-2026	Date of Acceptance: 20-02-2026	Date of Publication: 10-03-2026

Abstract— The transition of the legacy database to the cloud environment is one of the key factors for achieving scalability, flexibility, cost efficiency, and data innovation for an organization. One of the most complicated processes in database migration involves the transformation of schemas, meaning their adaptation from the format of the legacy database to a cloud environment. The conventional methods for schema transformation heavily rely on manual work and mapping rules, thus making the process resource-consuming and prone to human errors. The ever-growing complexity of enterprise databases exacerbates these problems, especially in case of heterogeneous systems, unknown schema structure, or constantly changing requirements. Thanks to the recent progress in Artificial Intelligence (AI), there are many opportunities for automating the process of schema transformation within the legacy-to-cloud migration framework. Such methods utilize machine learning, semantic analysis, pattern recognition, knowledge graphs, and language models to identify correspondences between schemas, infer relationships between different data sets, provide suggestions about transformation rules, and create scripts for migration without any human assistance. Such technologies can greatly improve the process of transformation, increase migration accuracy, reduce transformation efforts and facilitate cloud migration. This paper presents an overview of existing literature on AI-assisted schema transformation and automation of the migration process. According to the findings, hybrid approaches, combining traditional matching techniques with advanced AI-assisted semantic analysis, demonstrate greater effectiveness and accuracy of transformation.

Keywords— Legacy Database Migration , Cloud Database Migration , Schema Transformation , Artificial Intelligence (AI) , Schema Matching , Automated Data Migration , Machine Learning

INTRODUCTION

The quick emergence of cloud computing has completely transformed the way enterprises handle data. Most of them adopt cloud databases because of the improved levels of scalability, elasticity, high availability, and cost savings. The cloud-based database solution offers automatic management, resource allocation, disaster recovery services, and easy integration with analytics and AI solutions. Hence, enterprises are working towards moving their legacy database systems to cloud databases to improve their agility and digital transformation efforts.

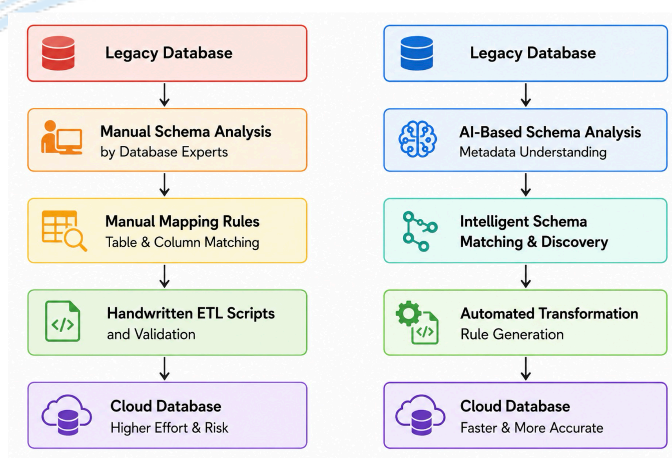

Fig. 1: Traditional Schema Migration vs AI-Assisted Migration

While cloud migration is beneficial for businesses, it poses several challenges, especially when enterprises need to move their legacy database systems. This is due to the fact that most of these legacy databases have been in use for several years now, with complex structures, data storage methods, unknown dependencies, and built-in business rules among others. These databases support crucial applications, which means any errors during migration could lead to serious problems. Hence, enterprises should be careful about migrating their database system.

Another critical issue related to legacy database migration to the cloud is schema transformation. It refers to the process of conversion of legacy source database schemas into target database schemas in compliance with specific cloud computing requirements and ensuring semantic consistency and business rules preservation. In many cases, legacy databases tend to use different naming convention schemes, data types, indexing, normalization, etc., than those provided by modern cloud environments. Traditional migration methods are mainly based on manual schema analysis, rules, and expert work.

Thanks to recent progress in artificial intelligence (AI), new possibilities are emerging for automating the schema transformation process. The idea behind AI-assisted migration is the usage of such techniques as machine learning, natural language processing, semantic matching, pattern recognition, and knowledge-based inference. They allow automatically detecting the connection between schema pairs. Modern AI technologies are capable of analyzing the schema metadata, finding attribute correspondence, generating transformation rules, anomaly detection, as well as generating migration scripts. This makes the migration process easier, more accurate, and consistent.

With the development of complex enterprise data architectures and increasing requirements for cloud modernization, there is currently a great research need in AI-assisted schema transformation. It is possible to say that AI-assisted migration combines all the best from both database engineering and intelligent automation domains.

Literature Review

Legacy-to-cloud database migration goes beyond simple data transfer between two database systems. In many cases, it involves intelligent understanding of schemas, automated schema matching, semantic mapping, data-type conversion, dependency analysis, and other steps. Schema problems typical for legacy databases include lack of documentation, duplicates, inconsistency in field names, embedded business rules, stored procedures, platform restrictions, and more. All these factors make manual migration difficult and expensive, especially at large scale. Intelligent schema transformation is an area where artificial intelligence can help. Rule-based matching, machine learning, semantic similarity, neural networks, and now also large language models become common AI techniques for identifying correspondences between schemas.

Some of the earliest attempts to automatize the schema matching problem laid the foundations of AI-assisted migration. For example, in one of the key surveys about schema matching Rahm and Bernstein (2001) described several groups of schema matching algorithms, including schema-level, instance-level, element-level, structure-level, linguistic, constraint-based, and hybrid techniques. This study is important because legacy-to-cloud migration implies exactly those capabilities to understand that different field names describe the same concepts and sometimes different-looking schemas should be transformed before being copied.

Next milestone is Cupid - a schema matching algorithm proposed by Madhavan, Bernstein, and Rahm (2001). This approach used linguistic similarity, data types comparison, and structure matching. This solution becomes crucial for automation of cloud migration, proving the idea that successful schema matching cannot rely on column names only. Moreover, constraints, hierarchies, and relationships should also be taken into account. Similarly, Li and Clifton suggested a neural network-based schema matching approach called SEMINT, showing that machine learning can help in interpreting schemas before the era of AI.

The COMA system developed by Do and Rahm (2002) added an innovative concept of a modular approach to schema matching, allowing combining multiple matchers to build the final schema correspondence. This idea proves its value for automated legacy-to-cloud migration when any single matcher is not enough for a comprehensive system transformation. Indeed, while some legacy tables will be matched based on field names, others may require constraint or instance-level matching, and special domain knowledge may be needed for some business fields. Later, COMA++ was proposed as an extension that supports matching schemas to ontologies.

Schema matching research also gave birth to automated schema transformation techniques. According to Miller, Haas, and Hernández (2000) schema mapping is actually a process of query discovery that generates queries for transforming source data to a desired target schema. This concept explains why transformation is a much more complicated step that just matching columns. Indeed, joins, splits, aggregations, normalization, denormalization, and data conversions have to be done for transformation to complete successfully. This topic was also discussed by Fagin et al. (2009) in Clio project that demonstrates semi-automation of schema mapping and transformation.

All research about automated cloud migration emphasizes the importance of automatization of schema transformation in

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 1, Jan – Mar 2026 || PP. 50 – 57 https://sjaibt.org

migration processes. Mehra (2014) proposes an automatic model for data migration that transfers a relational database to cloud environment. The model shows that automation is crucial for cloud migration when relational structures and format of the target database should be carefully managed. Also, Alcañiz et al. (2014) investigated security issues related to legacy-to-cloud data migration and explained that security considerations go beyond technical details and involve preserving confidentiality, integrity, access control, and compliance. Hasan et al. (2023) conducted a systematic literature review on cloud migration that includes issues related to pre-migration, migration itself, and post-migration.

Nowadays, researches about automated large-scale schema matching and data integration also contribute to migration. For example, in the study by Bernstein, Madhavan, and Rahm (2011) authors reconsidered the problem of generic schema matching ten years later and concluded that problems remain due to semantic ambiguity, large volume of data involved, and contextual nature of schema matching. Koutras et al. (2021) introduced a novel schema matching evaluation framework, called Valentine, that tests schema matching techniques in different scenarios including unionable and joinable datasets which resemble cloud migration tasks involving migration of various legacy databases into the cloud.

Automated cloud migration can benefit from deep learning technologies. For example, Mudgal et al. (2018) applied deep learning for entity matching task, proving that neural networks improve record matching and linking. This approach does not match exactly to schema matching. However, it can assist automated cloud migration by helping detect duplicated records or check referential integrity after migration is finished. Additionally, Li et al. (2020) introduced Ditto, a deep learning-based framework for entity matching and data integration. They showed that transformers can assist in solving semantic matching tasks in automated data transformation.

Last but not least, AI can now apply large language models (LLMs) in schema matching task. Parciak et al. (2024) conducted experiments using LLMs for schema matching and demonstrated that LLMs can facilitate the bootstrapping process using only schema names and description, although quality of the result greatly depends on the context preparation process. The findings are valuable for cloud migration considering that many legacy systems do not have proper documentation, and access to production data is not available for security or privacy reasons. LLMs can help analyze and interpret field names, detect schema correspondences, explain transformation process, and produce migration scripts. However, results of this kind still need human verification and validation.

Objectives and Research Methodology

The increasing complexity of enterprise database environments has created a strong demand for intelligent solutions that can automate schema transformation during legacy-to-cloud migration. Traditional migration approaches often require extensive manual intervention for schema analysis, attribute mapping, constraint identification, and transformation rule generation. These activities increase migration costs and project timelines while introducing the risk of inconsistencies and human errors. The primary objective of this study is to investigate the role of Artificial Intelligence (AI) in automating schema transformation and improving the efficiency of cloud database migration processes.

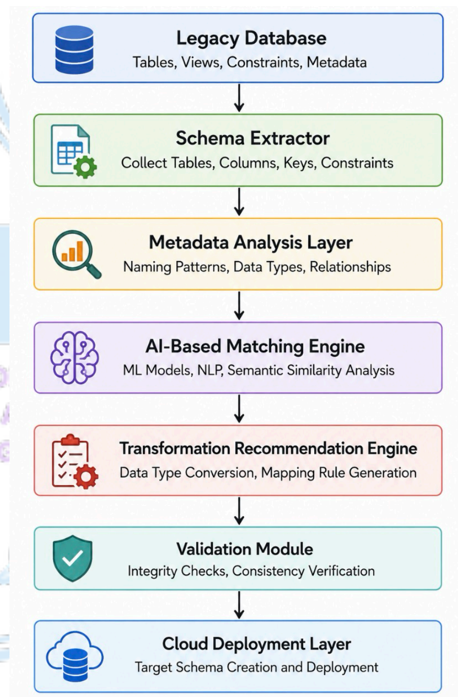

Fig. 2: Proposed AI-Assisted Transformation Framework

Firstly, there is a need to implement automated schema matching using artificial intelligence approaches such as machine learning, semantic analysis, and pattern recognition. Automated schema mapping allows detecting similarities between structures of both source and target databases without conducting extensive configuration work. Secondly, there is a necessity to make database migration processes more efficient through decreasing the amount of human interventions needed for analysis and transformation of database schemas. Thirdly, one should ensure that all elements of source data schemas (structures, constraints, and business rules) are properly transferred and implemented in target cloud schema. Artificial intelligence solutions allow reducing errors in mapping data from legacy database and improving semantic consistency.

Finally, another important goal is to accelerate the adoption of cloud technologies by making migration processes shorter and more reliable.

To realize the mentioned goals, a qualitative review research design was chosen as the main research method. The relevant scholarly literature, peer-reviewed journal articles, and other academic sources focusing on the areas of schema matching, database migration, modernization of cloud architecture, machine learning, and artificial intelligence solutions for data integration have been carefully studied and analyzed. This literature has been reviewed based on four major criteria – automation, transformation accuracy, migration efficiency, and reliability. Comparative analysis of the existing literature allowed identifying typical techniques used for schema mapping and transforming, their benefits and limitations, and new directions in the field. Key performance indicators used in the literature include time needed to transform data and migration duration among others.

Table 1: Research Objectives and Evaluation Metrics

Objective	Evaluation Metric
Automation	Transformation Time
Accuracy	Schema Mapping Accuracy
Efficiency	Migration Duration
Reliability	Error Rate

LEGACY-TO-CLOUD MIGRATION CHALLENGES

Migration from legacy database solutions to cloud-based DBMS is a very complex process, which comprises much more than transferring of data between two environments. The main reason for such problems appears to be incompatibility of various components between legacy database infrastructure and modern cloud database platforms. Therefore, addressing all potential issues is key to performing the migration successfully.

First of all, the problem of heterogeneous databases occurs often enough. Enterprise-level organizations often have several database technologies accumulated for many years, which could be of both hierarchical, object, and relational varieties. Moreover, each database technology will have its own structure of schema, language, storage mechanism, and metadata.

Another significant issue is the one of schema incompatibility. Legacy DBMS are known to contain many nested schemas, specific naming policies, undocumented relations between schema elements, redundant attributes, as well as different approaches towards schema organization. Therefore, mapping such schemas to cloud-native structures would be quite difficult.

Also, the conversion of data types should not be overlooked. Different DBMS have their own set of supported data types, along with various storage, indexing, and encoding mechanisms, which could differ greatly. Therefore, there might appear cases when a specific data type is present in the source legacy DBMS but cannot be found in the target cloud DBMS.

Moreover, migration could also suffer because of the mapping of constraints. Data integrity in legacy database solutions relies on such tools as primary and foreign keys, unique constraints, triggers, indexes, stored procedures, and even business rules implemented within the database itself. All these constraints will have to be properly transformed to fit into the new cloud database environment.

Finally, performance is an important factor to consider when migrating a database solution to the cloud. Enterprise-level databases usually include a very large number of data records (millions or more), as well as a heavily interconnected database structure. Migration operations may have negative effects on database performance and overall application stability, and therefore need to be handled carefully.

All of these challenges make migration process rather challenging and complicated. However, due to the development of intelligent automation and AI-based tools, many of them could be solved relatively easily.

Table 2: Common Schema Migration Challenges

Challenge	Impact
Data Type Mismatch	Conversion errors
Schema Complexity	Increased migration effort
Constraint Differences	Integrity issues
Vendor-Specific Features	Compatibility problems

AI-ASSISTED SCHEMA TRANSFORMATION FRAMEWORK

However, the growing complexity of legacy database systems makes it necessary to create intelligent frameworks that would help automate schema transformation during the process of cloud migration. Manual approaches that involve database specialists' analysis of source schema, mapping definition, rule creation, and testing require substantial time and effort when dealing with large enterprise databases with thousands of tables, attributes, relations, and constraints. Therefore, it would be reasonable to employ an AI-enabled framework designed to assist with automation of the entire process of schema transformation.

To begin with, an AI-enabled schema transformation framework starts with schema extraction layer. Its main purpose is to extract metadata from the legacy system including all information regarding table names, column definitions, PKs,

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 1, Jan – Mar 2026 || PP. 50 – 57 https://sjaibt.org

FKs, indexes, constraints, stored procedures, views, and relationships. All the obtained data will form the knowledge base for further analysis. By using this method, the amount of effort required to understand large legacy systems will be greatly reduced.

Secondly, a Metadata Analysis Module should be developed. Its goal is to perform preliminary analysis and normalize gathered data related to source schemas. In other words, naming patterns, attribute types, relation structures, and usage frequency will be analyzed. Moreover, metadata can be enhanced using business glossaries, document repositories, and database history records. The main objective is to build a normalized model of the source schema.

The most important element of the suggested framework is represented by an AI-based schema matching engine. Machine learning models, semantic analysis, NLP technologies, pattern recognition, and knowledge-based approaches will be used to identify corresponding elements between source and target schemas. As opposed to existing tools that are primarily focused on name matching, this engine will focus on analysis of relationships, meanings, properties, and structure dependencies. It means that AI models can recognize equivalence even if naming standards differ substantially.

After matching is performed, an automatic Transformation Recommendation Engine will create migration rules for each mapped entity. Depending on analysis results, actions that include conversion, table reconstruction, normalization or denormalization, key conversion, creation of indexes, etc., will be chosen. According to established relations between different objects, transformation rules will be produced that can be directly executed on a target platform.

At this stage, a Validation Module should check the correctness and completeness of the created mappings. The validation procedure includes verification of consistency, maintenance of relationships and constraints, preservation of referential integrity, and adherence to business rules. With the use of AI-assisted anomaly detection methods, errors that may occur during the process of transformation will be revealed.

Finally, a Cloud Deployment Layer will be responsible for applying created schema to the target database system. Using created transformation rules, the new database structure will be constructed including all constraints, indexes, etc. Additionally, monitoring functions can be added to ensure successful deployment.

In summary, this approach can be regarded as a combination of metadata-driven analysis and artificial intelligence technology that allows automating and scaling up the process of schema transformation.

Table 3: Framework Components

Component	Function
Schema Extractor	Collects metadata
AI Matching Engine	Identifies schema mappings
Transformation Generator	Creates migration rules
Validation Module	Verifies schema consistency
Cloud Deployment Layer	Applies transformed schema

INTELLIGENT SCHEMA MAPPING AND TRANSFORMATION PROCESS

The most important steps in the described migration framework are intelligent schema mapping and transformation. They refer to the accurate mapping of legacy database structures to new compatible schemas along with preserving all business logic and database consistency. While traditional approaches use a lot of manual work and custom coding, modern intelligent transformation systems allow automatic mapping and converting of schemas and database content.

The first step to be performed includes column matching when the columns of the source database are analyzed in order to find matches among the target schema objects. Using advanced machine learning and deep learning algorithms, an intelligent engine detects semantically related columns, regardless of their names and differences in naming practices. For instance, Cust_ID column of a legacy database can be correctly detected by an intelligent system as CustomerIdentifier in the target cloud schema. Intelligent column matching is very useful since it allows significantly reducing manual effort during the migration process.

The next step to perform consists in entity-relationship mapping when the complex inter-table relations in the source legacy database are being analyzed and preserved during migration. There are one-to-one, one-to-many, and many-to-many relationships between tables, which should be discovered and mapped using intelligent algorithms. This step is necessary to ensure relational integrity of the migrated database.

The next step of the transformation framework is related to data type conversions. Different databases have their own sets of data types, their precision and storage capabilities. It is crucial to analyze the compatibility between source and target platforms, and then convert some specific types into universal cloud-supported data types. The automation of this step saves time and eliminates manual errors.

Constraint transformation is another important stage to perform. The constraints include primary keys, foreign keys, unique constraints, check constraints, indexes, and triggers. It is important to transfer all integrity rules into the target cloud database. An intelligent engine will help to analyze all these items and offer adequate transformations preserving consistency.

At last, automated code generation is an important step to perform. The transformation engine analyses all mappings between legacy and target schema, and generates the migration scripts in a target format. This step is very useful because migration code generation allows fast execution, reduced effort, minimized risk and error rate.

Table 4: Example Schema Transformation Rules

Legacy Schema Element	Cloud Schema Equivalent
VARCHAR2	VARCHAR
NUMBER	DECIMAL
Stored Procedure	Cloud Function
Trigger	Event-Based Workflow

Conceptual Comparative Evaluation

To maintain consistency with the qualitative review methodology adopted in this study, this section presents a conceptual comparative evaluation synthesized from trends reported in the reviewed literature rather than results obtained from a controlled experimental implementation. The purpose of this evaluation is to illustrate how AI-assisted schema transformation frameworks are expected to perform relative to traditional migration approaches based on observations and conclusions reported by prior research in schema matching, data integration, and cloud migration.

A representative migration scenario can be conceptualized using enterprise legacy database environments that commonly include platforms such as Oracle Database, Microsoft SQL Server, and MySQL. Likewise, modern cloud migration initiatives frequently target platforms such as Snowflake, PostgreSQL, and BigQuery because of their scalability, cloud-native capabilities, and analytical features. The reviewed literature suggests that schema transformation between such heterogeneous environments typically requires schema matching, data-type conversion, constraint preservation, dependency analysis, and transformation-rule generation.

Contemporary AI-assisted migration frameworks commonly employ machine learning techniques, semantic similarity analysis, metadata interpretation, natural language processing, knowledge-based reasoning, and, increasingly, large language models to support schema transformation activities. Rather than relying exclusively on predefined mapping rules, these approaches attempt to identify semantic relationships between schema elements by analyzing metadata, naming conventions, structural dependencies, constraints, and contextual information. The resulting recommendations can then support migration specialists during schema transformation and validation activities.

The comparative assessment presented in this section focuses on three evaluation dimensions frequently discussed in the literature: schema-mapping quality, migration effort and duration, and transformation reliability. Previous studies generally suggest that AI-assisted approaches may provide advantages over purely manual and rule-based techniques by improving semantic interpretation, reducing repetitive mapping activities, and supporting validation of transformation outputs. However, the degree of improvement depends on factors such as metadata quality, schema complexity, domain knowledge availability, and the level of human oversight applied during migration projects.

The qualitative ratings presented in the following tables are intended solely to summarize recurring patterns reported across the reviewed literature. They should not be interpreted as the outcome of a specific experimental deployment, benchmark study, or production migration environment. Instead, they provide a conceptual representation of the relative strengths and limitations of alternative schema-transformation approaches discussed throughout the literature.

Table 5: Conceptual Evaluation Basis

Parameter	Value
Evaluation Type	Literature-based conceptual comparison
Source Context	Legacy relational database schemas
Target Context	Cloud database schema transformation
AI Approach	AI-assisted semantic/schema matching
Evaluation Focus	Accuracy, time, and error reduction

Table 6: Schema Mapping Accuracy — Conceptual Comparison

Method	Expected Mapping Accuracy
Manual Schema Mapping	Moderate
Rule-Based Mapping	Moderate to High
Metadata Matching Tool	High
AI-Assisted Framework	High, subject to validation

Table 7: Migration Time Comparison

Method	Expected Migration Time
Manual Migration	High
Rule-Based Migration	Medium
Automated Mapping Tool	Medium to Low
AI-Assisted Framework	Lower, with human validation

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 1, Jan – Mar 2026 || PP. 50 – 57 https://sjaibt.org

Table 8: Error Rate Comparison

Method	Expected Error Risk
Manual Migration	High
Rule-Based Migration	Medium
Automated Mapping Tool	Lower
AI-Assisted Framework	Lower, if validated

FINDINGS AND DISCUSSION

The review of existing literature indicates that artificial intelligence has considerable potential to support schema transformation during legacy-to-cloud database migration. Across the studies examined, AI-assisted approaches consistently emerge as promising mechanisms for addressing the limitations of traditional manual and rule-based migration techniques, particularly in environments characterized by complex schemas, heterogeneous database platforms, and incomplete documentation.

One of the most frequently reported observations in the literature concerns the ability of AI-assisted schema matching techniques to improve the identification of semantic relationships between source and target database elements. Traditional migration methods often depend on manual interpretation of schema structures or predefined matching rules that may struggle to capture contextual meaning when naming conventions differ across systems. In contrast, machine learning, semantic analysis, and natural language processing techniques can evaluate multiple sources of information, including metadata, structural relationships, attribute characteristics, and schema descriptions. As a result, AI-assisted approaches may provide more comprehensive support for schema correspondence discovery in complex migration scenarios.

The reviewed studies also suggest that intelligent automation can reduce the effort associated with migration planning and schema transformation. Activities such as metadata analysis, schema interpretation, mapping preparation, transformation-rule generation, and validation often require substantial manual involvement when performed using conventional approaches. AI-assisted systems may help streamline these activities by generating recommendations and identifying potential mappings automatically, allowing migration teams to focus on review, validation, and business-rule verification rather than repetitive analytical tasks.

Another recurring theme in the literature is the role of AI in improving migration reliability. Several studies emphasize that automated validation mechanisms, anomaly detection techniques, and semantic consistency checks can assist in identifying potential transformation issues before deployment. Such capabilities may help reduce the likelihood of mapping inconsistencies, unsupported transformations, referential-integrity violations, and schema compatibility problems. However, the literature also emphasizes that AI-generated recommendations should be subject to expert review, particularly in mission-critical environments where business logic, regulatory requirements, and domain-specific constraints must be preserved accurately.

The review further indicates that hybrid approaches combining traditional schema-matching techniques with AI-driven analysis appear to be more practical than fully automated solutions. Existing research suggests that successful schema transformation often requires a combination of linguistic matching, structural analysis, metadata interpretation, domain knowledge, and human validation. Consequently, AI is best viewed as an intelligent decision-support mechanism that augments rather than replaces database architects and migration specialists.

Overall, the literature suggests that AI-assisted schema transformation can contribute to improved migration efficiency, greater consistency in schema mapping, reduced manual effort, and enhanced support for migration validation activities. Nevertheless, the effectiveness of these approaches remains dependent on factors such as metadata quality, schema complexity, availability of contextual information, and the extent of expert oversight applied throughout the migration process. Because the present study is based on a qualitative review of prior research, the proposed framework should be regarded as a conceptual synthesis of current knowledge. Future empirical studies are needed to evaluate its performance under controlled and reproducible migration scenarios involving real-world legacy and cloud database environments.

CONCLUSION AND FUTURE WORK

Legacy-to-cloud database migration remains a complex undertaking because of schema heterogeneity, incompatible data types, undocumented relationships, embedded business rules, and platform-specific implementation details. Among the various migration activities, schema transformation represents one of the most critical and resource-intensive tasks, requiring accurate identification of correspondences between source and target database structures while preserving semantic meaning and data integrity. As organizations increasingly modernize their data infrastructures, the need for more intelligent and scalable approaches to schema transformation continues to grow.

This study reviewed the existing literature on schema matching, schema mapping, automated database transformation, cloud migration, and artificial intelligence applications in data integration. The analysis indicates that AI-assisted approaches can provide valuable support for schema transformation by combining semantic analysis, metadata interpretation, machine learning, pattern recognition, and automated recommendation mechanisms. Compared with purely manual or rule-based approaches, AI-assisted methods appear better suited for handling complex schemas, identifying semantic relationships, and supporting transformation planning in heterogeneous database environments.

The review further suggests that hybrid approaches combining traditional schema-matching techniques with AI-driven analysis may offer the most practical solution for enterprise migration projects. Rather than replacing human expertise, AI can function as a decision-support mechanism that assists database architects in identifying mappings, generating transformation recommendations, validating schema consistency, and reducing manual effort. Such collaboration between human expertise and intelligent automation has the potential to improve migration quality, consistency, and operational efficiency.

However, the present study is based on a qualitative review of existing literature and proposes a conceptual framework rather than reporting results from a controlled experimental implementation. Consequently, the proposed framework should be regarded as a theoretical model that synthesizes current research trends and identifies opportunities for future development.

Future research should focus on empirical validation of AI-assisted schema transformation frameworks using real-world migration datasets, clearly defined evaluation procedures, and reproducible experimental settings. Additional investigation is also needed into large language model–based schema interpretation, explainable AI techniques for schema matching, automated generation of migration rules and scripts, and intelligent validation mechanisms capable of supporting large-scale multi-cloud migration environments. These developments may contribute to the creation of more autonomous, reliable, and scalable database migration solutions capable of supporting the next generation of cloud modernization initiatives.

References

- Alcañiz, L. M., Gómez, A., & Marcos, E. (2014). Security in legacy systems migration to the cloud. *Proceedings of the International Conference on Cloud Computing and Services Science*.
- Aumüller, D., Do, H. H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with COMA++. *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 906–908. https://doi.org/10.1145/1066157.1066283
- Bernstein, P. A., Madhavan, J., & Rahm, E. (2011). Generic schema matching, ten years later. *Proceedings of the VLDB Endowment, 4*(11), 695–701. https://doi.org/10.14778/3402707.3402710
- Do, H. H., & Rahm, E. (2002). COMA: A system for flexible combination of schema matching approaches. *Proceedings of the 28th International Conference on Very Large Data Bases*, 610–621.
- Fagin, R., Haas, L. M., Hernández, M., Miller, R. J., Popa, L., & Velegrakis, Y. (2009). Clio: Schema mapping creation and data exchange. In *Conceptual Modeling: Foundations and Applications* (LNCS 5600, pp. 198–236). Springer. https://doi.org/10.1007/978-3-642-02463-4_12
- Hasan, M. H., Osman, M. H., Admodisastro, N. I., & Muhammad, M. S. (2023). Legacy systems to cloud migration: A review from the architectural perspective. *Journal of Systems and Software, 202*, 111702. https://doi.org/10.1016/j.jss.2023.111702
- Koutras, C., Siachamis, G., Ionescu, A., Psarakis, K., Brons, J., Fragkoulis, M., Lofi, C., Bonifati, A., & Katsifodimos, A. (2021). Valentine: Evaluating matching techniques for dataset discovery. *Proceedings of the IEEE International Conference on Data Engineering*.
- Li, W. S., & Clifton, C. (2000). SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering, 33*(1), 49–84. https://doi.org/10.1016/S0169-023X(99)00044-0
- Madhavan, J., Bernstein, P. A., & Rahm, E. (2001). Generic schema matching with Cupid. *Proceedings of the 27th International Conference on Very Large Data Bases*, 49–58.
- Mehra, K. (2014). *Automatic data migration into the cloud* [Master's thesis, Concordia University].
- Miller, R. J., Haas, L. M., & Hernández, M. A. (2000). Schema mapping as query discovery. *Proceedings of the 26th International Conference on Very Large Data Bases*, 77–88.
- Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., & Raghavendra, V. (2018). Deep learning for entity matching: A design space exploration. *Proceedings of the 2018 International Conference on Management of Data*, 19–34. https://doi.org/10.1145/3183713.3196926
- Parciak, M., Vandevoort, B., Neven, F., Peeters, L. M., & Vansummeren, S. (2024). Schema matching with large language models: An experimental study. *VLDB Workshop on Tabular Data Analysis*.