Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 20(4): 265-272, Dec. 2022 Regular paper

Relational Data Extraction and Transformation: A Study to Enhance Information Systems Performance

Forat Falih Hasan[1,2]* and Muhamad Shahbani Abu Bakar[1], *Member, KIICE*

[1]School of Computing (SOC), Universiti Utara Malaysia (UUM), Sintok, Kedah 06010, Malaysia
[2]Department of Computer Engineering Techniques, College of Engineering Technology, Alkitab University, Altun Kupri 36001, Iraq

Abstract

The most effective method to improve information system capabilities is to enable instant access to several relational database sources and transform data with a logical structure into multiple target relational databases. There are numerous data transformation tools available; however, they typically contain fixed procedures that cannot be changed by the user, making it impossible to fulfill the near-real-time data transformation requirements. Furthermore, some tools cannot build object references or alter attribute constraints. There are various situations in which tool changes in data type cause conflicts and difficulties with data quality while transforming between the two systems. The R-programming language was extensively used throughout this study, and several different relational database structures were utilized to complete the proposed study. Experiments showed that the developed study can improve the performance of information systems by interacting with and exchanging data with various relational databases. The study addresses data quality issues, particularly the completeness and integrity dimensions of the data transformation processes.

Index Terms: Database, Data transformation, R-programming, Extract transform and load, Information systems

I. INTRODUCTION

Most organizations use relational database management systems (RDBMSs) to store and manage data in data storage systems [1]. This system is based on a relational model and employs table-and-relationship concepts. The RDBMS was created using the structured query language (SQL) to facilitate various operations in relational databases of all types [2,3]. The field of data transformation has recently attracted considerable attention. Because data with logical structures are transformed from one system to another, the processes for this activity must be well-specified and based on sound concepts. Therefore, metadata comprises information about the data or describes other data in relational databases. It is used to transfer data across different information systems (ISs) and share the required data. Thus, the processing of other data is known as metadata management. It includes comprehensive details on databases, such as views, tables, columns, users, structures, and processes [4-6].

The process of converting data from one format to another is known as data transformation. Extract, transform, and load (ETL) is a general term for moving the data from one system to another [7,8]. In the ETL technology, there are two primary terms for data storage: data sources and target systems. The target system is used to load (process) the data after they are extracted from data sources [9-11]. In ETL, the complete data extraction from the sources is referred to as complete-extraction, whereas partial extraction divides the extraction process into two distinct phases. As part of the transformation process, many functions are applied to the

Received 16 October 2021, **Revised** 25 November 2022, **Accepted** 28 November 2022
***Corresponding Authors** Forat Falih Hasan[1,2] (E-mail: forat.db@gmail.com,forat@uoalkitab.edu.iq), Muhamad Shahbani Abu Bakar[1] (E-mail: shahbani@uum.edu.my)
[1]School of Computing (SOC), Universiti Utara Malaysia (UUM), Sintok, Kedah 06010, Malaysia
[2]Department of Computer Engineering Techniques, College of Engineering Technology, Alkitab University, Altun Kupri 36001, Iraq

Open Access https://doi.org/10.56977/jicce.2022.20.4.265 print ISSN: 2234-8255 online ISSN: 2234-8883

retrieved data. In contrast to other types of data, direct transfers do not have to be processed or altered. Moreover, when data are loaded for the first time, the complete load approach is employed, involving the complete cleaning of tables. In addition, when adding modified or new records to a target source, an incremental load is employed [12-15].

The relational information systems of an organization may rely on more than one RDBMS, such as Oracle for human resources, MySQL for billing systems, and SQL Server for other activities, as is normal practice in any organization. For query construction and decision making based on data from multiple sources, considerable effort is required. This study aims to overcome data quality issues in transformation procedures by transforming data with metadata between different relational databases and offering uniform data access and sharing. Data quality is considered the main factor that determines whether transformation data processing is successful [16,17]. In relational databases, quality issues arise if the source and target metadata data are dissimilar. In addition, non-original metadata causes variations in the logical structures of source and target systems. During data transformation, many quality issues arise between the source and target systems, such as data type mismatch, key constraint mismatch, attribute constraint mismatch, attribute name mismatch, removal of referential integrity constraints, missing values, duplicate records, wrong data mapping, table name mismatch, data size mismatch, and missing values. These scenarios fall into data completeness and integrity [18,19].

The primary benefits of the developed method are as follows: First, it is easier to upgrade all relational databases to a new version of the same system. Second, it offers a consistent method for interacting with various RDBMSs and delivering the desired results. As a result of the developed study, fewer apps are required to handle and administer each type of database. Finally, by retaining the original metadata and resolving concerns regarding data quality, information systems may improve the performance of their data transformation.

II. RELATED WORK

Many relational database data transformation solutions have problems and limitations when transforming foreign keys previously utilized to establish links between database objects. These tools cannot be adjusted in real time and have limited flexibility in modifying and analyzing the data stored in the source and target databases. Many solutions cannot provide a statistical perspective on connected databases on both sides [7-20].

The study in [7] supports data transformation between various types of relational databases. First, the sources' database metadata are analyzed, and the logical structure with references is subsequently loaded. This method is suitable for transforming RDBMS with a small number of connected tables. With no permission to edit the tables before or after the transformation, the data in the sources and target databases cannot be compared.

The technique based on XML, Java, and Oracle improves ETL metadata management [21]. The operational data are stored in a table with limitless columns. Users must define and create the structure of the required tables in the Oracle database and then run the programs to modify the data for the created tables. This approach only supports partial RDBMS transformation of a single table to target sources and not the entire database. In addition, users must define the repository structure of the converted tables. Thus, [22] improved Hyper-E-T-L by increasing the processing speed. AE-T-L divides tables into pieces and then runs the transformation process. This method is helpful if only a table is required to be converted. The second approach adds multi-data-source interactions. Furthermore, [23] developed an open-source program that could interact with multiple inputs and read stored data. The developed framework E-T-L-ator is based on Python and includes three primary levels: connectors, loggers, and tables. The main concept of the connecter class is to provide a proper connection with multiple relational databases. Furthermore, the table class contains all the information from the linked databases. In this study, only one table can be transformed from the source to the target system, and the entire database data and objects cannot be transformed. In addition, the user must create the required structure of the table in the target destination and then transform the data based on the predefined process; there is no flexibility in auto-creating the data structures in the target sources. All the structures must be defined and generated manually. [24] presented study based on the concepts of transforming the relational database into a data warehouse repository; the first process starts by creating the structure of the required design in the target part and, then transforming the data from one source only. Based on this study, all table structures must be created by users, and the data must flow from a single database. [20] developed a method based on three main concepts. It effectively manages metadata and adds adds intelligent tools to enhance decision making. In this method, the database structure is manually created on the target side by the user. The method developed in [25] comprises three main parts. The first extracts the data from the sources, cleans all the dirty data, and stores them in the middle library. The middle library receives queries from the third part, T and L, to inject the required data into the target database. This method secures the transformation for only one table and does not permit the transformation of the entire database. There is no permission to transform all database tables with P-K and F-K, and the structure of the target database is already created by the user to receive the cleaned data. Thus, [8] presented a method for transforming data stored in Excel sources into a relational database. This method was based on web processing, which could extract, transform, and load only one table from the sources.

All the earlier techniques presupposed that the source table was converted into the destination database and that all data types were supported by the attributes, which had been built as text data types. A database had to be assigned to each method, and it could not serve as both the source and destination of the data flow. A single primary key column was assumed for all solutions in each table. Data quality issues arise from discrepancies in metadata between the source and destination systems.

III. RESEARCH APPROACH

The R programming language was used in this study. The approach integrates Re-DEM, which is used to extract data from relational databases, with Re-DTM, which is used to transform that data. Adding a central library environment (CLE) to ETL operations facilitates the management and organization of data flow between various relational databases in the source and target systems. Five main processes were involved: source RDBMSs, Re-DEM, CLE, Re-DTM, and destination RDBMSs. The initial step of the source system involved communicating with several RDBMS and serving as the input for the next operation. Moreover, Re-DEM was employed in the second step to interact with the source RDBMSs and load the outcomes into the CLE. Third, data from source systems were imported into the CLE along with many sublibraries that included the extracted logical structure information. Furthermore, based on the CLE data, the fourth step involved employing Re-DTM to load the final results and requirements into the target RDBMSs. Finally, the fifth step included the target system (RDBMS).

This study focused on determining methods to unify data transformations across RDBMSs. When interacting with various data sources, it is necessary to save and categorize the data obtained. CLE was developed to facilitate high-quality, flexible data transformation across various RDBMSs. In addition, all databases are connected to the R environment; at this level, the source and target databases were combined in one environment. This method allowed flexible and near-real-time RDBMS data transformation.

The Re-DEM method architecture used ETL-based algorithms to identify available RDBMSs in the source system, then analyze each RDBMS and gather related data. Therefore, specific metadata-based algorithms have been developed for each RDBMS to link, read, analyze, and transform stored data with logical structures to CLE. The connection name for each database was used by the Re-DEM in the extraction process to identify the linked RDBMS types. The next step involved determining the metadata structure and data stores, as well as the rules required to interpret and extract the logical metadata structure. This includes attribute properties, such as attribute names and data types. The direct move applies to all data loaded from the source system to the CLE, and the data passed without modification. The loading process was based on the full load method [7], which was used to insert data for the first time. All three types of transformation operations were used in this step for this study [22].

The CLE served as a bridge between the two processes of extracting and transforming relational data. All data were saved in CLE using the RStudio environment. Furthermore, each RDBMS used sublibraries to organize data from a single source.

The Re-DTM operations and processes are based on the Re-DEM results that are stored in the CLE. Each sublibrary comprises an RDBMS structure and data that can be injected into any target RDBMS. To maintain database integrity, relational databases require standardized and integrated data. To ensure that the related tables are correct, referential integrity must be appropriately passed on to the target RDBMS [26]. This method involves three stages: (a) converting the logical structure of each table, (b) filling each table with data on the target side, and (c) transforming the key constraints of each attribute by establishing relationships between the database tables.

IV. MODELLING OF THE DEVELOPED STUDY WITH RDBMS METADATA

The developed approach uses the source RDBMS metadata. This study explains how to access and retrieve MySQL metadata. Furthermore, the INFORMATION-SCHEMA in every MySQL instance stores details of all the database objects. There are multiple read-only views in this schema, indicating that no changes can be made to the data, only its presentation. This section of the database contains various tables related to operations [27], and all the data can be seen with SQL queries. In this section, the developed method is tested using the RDBMS, which was used as the source system and is based on the MySQL DB. Therefore, employee data and SQL scripts are the main items in this HR database Information-schema and information-key are the two metadata tables most often used in this study [28].

Algorithm-R1-1 for the MySQL connection (MSQL) based on the R-studio language is developed to set up the connection with the MySQL database and produces the connection name used to connect with the MySQL database in all of the other algorithms developed in this study. This algorithm serves as the basis for processes based on the Re-DEM. The flexibility to interact with one or more MySQL

DBs is provided by a specific library named R-MySQL [29]. To successfully finish this stage, this package is required to assign connection names, hostnames, usernames, and passwords. This provides a global connection name with the MySQL system, as shown below:

Algorithm-R1-1: MySQL Connection

```
Input: MySQL database
Output: Connection name with MySQL
Variables: connection_name, username, password, host
begin
    set "MSQL" is the connection name
    username ="root"
    password = "MySQL2020"
    host = "127.0.0.1"
    / * host based on the local connection */
    loading (DBI) library
    loading (RMySQL) library
    MSQL = starts connecting based on (username, password, host)
    / * "MYSQL" the output of current algorithm */
End
```

After establishing a connection between R-studio and MySQL, R2-1 was used to extract the logical structures (LgS) for all sorted tables. Using this method, all MySQL tables were read and saved to Library-1-2. Attributes with all characteristics were identified together with the logical structures of the linked RDBMS at this step. These details were saved in a specified place for each table property, as shown below.

Algorithm-R2-1: MySQL LgS

```
Input: MySQL metadata
Output: Library1-2 sub-library
Variables: isc,dbn
begin
    dbn = "database name"
    / * "dbn" Is the database name assigned by the users */
    isc = {all the tuples ∈ "information_schema.columns"}
    / * "information_schema.columns" Is the table name in MySQL metadata */
    for all the tuples isc
        read tuple isc
        if database name dbn then
            / * "isc" holds the information for all databases in MySQL instance */
            Insert isc tuple into Library-1-2 table
        next tuple in isc
end
```

To provide the RDBMS statistics information, Algorithm-R2-2 was developed after the Tables LgS were extracted. In addition, Library-1-1 lists the table names, properties, and row information. This stage was designed to offer a clear image of the database before and after transformation, and to evaluate the accuracy of the proposed transformation algorithms by examining the source and destination data, as shown below:

Algorithm-R2-2: MySQL (MSI)

```
Input: MySQL metadata
Output: Library-1-1
Variables: ist, isc
begin
    ist = {all the tuples ∈ "information_schema.tables"}
    / * "information_schema.tables" Is the table name in MySQL metadata */
    isc = {all the tuples ∈ "information_schema.columns"}
    / * "information_schema.columns" Is the table name in MySQL metadata */
    create table Library-1-1
    / * Library-1-1 consists of three attributes */
    for all the tuples ∈ ist
        for all the tuples isc
            / * ist = number of rows, isc = number of columns */
            read tuple ist
            read tuple isc
            insert (table_name, ist, isc) into Library-1-1 table
            / * insert table name with the number of rows and attributes into Library-1-1 */
        next tuple in ist
        next tuple in isc
end
```

Algorithm-R3-1 was proposed to extract all the information related to the attributes, constraints, and table references. Essentially, Algorithm-R3-1 reads the constraints for each attribute in entire database tables and saves the results in the sub-libraries, Library-1-3 and Library-1-4 using the MySQL INFORMATION-SCHEMA. KEY-COLUMN USAGE Table. As shown in the algorithm processes below:

Algorithm-R3-1: MySQL Constraints Information (MCI)

```
Input: MySQL metadata
Output: Library-1-3 and Library-1-4 sub-libraries
Variables: isk
begin
    isk = {all the tuples ∈ "information_schema.key_column_usage"}
    / * "information_schema.key_column_usage" Is the table name in MySQL metadata */
    create tables Library-1-3 and Library-1-4
    for all the tuples ∈ isk
        read tuple isk
        insert isk tuple into Library-1-3 and Library-1-4 tables
        / * insert attributes constraints with the links of each table into Library1-3 and Library-1-4 tables */
        next tuple in isk
end
```

After extracting the source system LgS and data to a predefined RDBMS, Re-DEM verifies the data quality. Algorithm-R4-1 compares the source RDBMS and CLE sub-libraries. First, we examined the original RDBMS metadata, then sub-libraries, and compared them. This algorithm provides a report on Re-DEM. The general processes in Algorithm-R4-1 are as follows:

Algorithm-R4-1: Data quality verification

```
Input: Sources database tables and sub-libraries in central library
Output: Data quality reports
begin
check the number of columns
check the number of rows
check the names of the columns
check the data type of the columns
check the number rows in each columns
check the number of null values in each columns
check the metadata information
if {
    no variances between the source and central
    update report "No Error"
    } else {
    update report "Error"
    }
end if
display data quality information
end
```

Notably, many sub-libraries are generated as a result of the Algorithms R2-2, R3-1, and R4-1, as shown in the tables below:

Table 1. Library-1-1- the first output of Algorithm-R2-2

No	Table.Name	N.Columns	N.Rows
1	countries	3	25
2	departments	4	27
3	employees	11	107
4	jobs	4	19
5	Job_history	5	10
6	locations	6	23
7	regions	2	4

Table 2. Library-1-3 the output of Algorithm-R3-1

Constraint. Name	Table.Name	Column.Name	Ordinal. position	Referenced. Table. name	Referenced. Column. Name
P.K	countries	country_id	1	NA	NA
P.K	departments	department_id	1	NA	NA
P.K	employees	employee_id	1	NA	NA
P.K	jobs	job_id	1	NA	NA
P.K	locations	location_id	1	NA	NA
P.K	regions	region_id	1	NA	NA

V. MODELLING OF THE DEVELOPED STUDY WITH RDBMS METADATA

The relational data are transformed based on the CLE. As aforementioned, each RDBMS structure was explained using sub-libraries. An automated RDBMS-to-RDBMS transformation was created using Re-DTM through four essential processes. The CLE and target RDBMS were linked using Algorithm-R-5. Thus, Algorithm-R-6 established attribute constraints with references and created the LgS in the target RDBMS. Algorithm-R-7 was used to load data into the target RDBMS, and Algorithm-R-8 was employed to verify the RDBMS data quality problems.

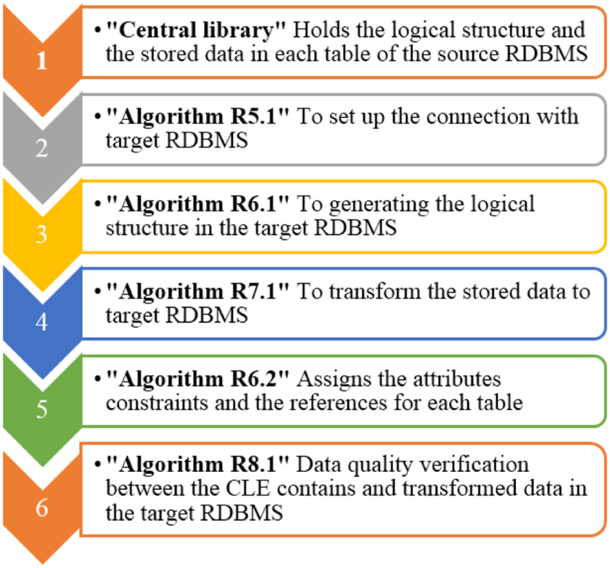
To describe the above explanation, consider Oracle as the destination RDBMS and then transform the CLE data logical structure to Oracle DB. Algorithm-R5-1 employed the same RDBMS connection processes as Algorithms R1-1 and R5-1 and output the connection name that was used in subsequent steps to connect to the target RDBMS. Moreover, Algorithm-R-6 generated logical RDBMS structures. In addition, the sub-library, Library-1-2 was transformed using Algorithm-

Table 3. Library-1-4- the second output of Algorithm-R3-1

Constraint.Name	Table.name	Column.name	Ordinal. position	Referenced Table.name	Referenced. Column.name
countries_ibfk_1	countries	region_id	1	regions	region_id
departments_ibfk_2	departments	manager_id	1	employees	employee_id
departments_ibfk_1	departments	location_id	1	locations	location_id
employees_ibfk_1	employees	job_id	1	jobs	job_id
employees_ibfk_3	employees	manager_id	1	employees	employee_id
employees_ibfk_2	employees	department_id	1	departments	department_id
job_history_ibfk_2	job_history	job_id	1	jobs	job_id
job_history_ibfk_1	job_history	employee_id	1	employees	employee_id
job_history_ibfk_3	job_history	department_id	1	departments	department_id
locations_ibfk_1	locations	country_id	1	countries	country_id

R6-1, and Algorithm-R7-1 identified the target RDBMS and its data type and format, read Library-1-2, converted it to the target format, and injected the produced LgS information into the target system.

This study was designed to transform database objects and store data without affecting content. The developed Algorithm-R7-1 read Library-1-1 of the source RDBMS in the CLE. It then inserted the data for each table separately using SQL codes inside the R environment. The full-load method was used because the data were loaded into the target database for the first time. The general processes under Re-DTM are presented in the figure below.



- **"Central library"** Holds the logical structure and the stored data in each table of the source RDBMS
- **"Algorithm R5.1"** To set up the connection with target RDBMS
- **"Algorithm R6.1"** To generating the logical structure in the target RDBMS
- **"Algorithm R7.1"** To transform the stored data to target RDBMS
- **"Algorithm R6.2"** Assigns the attributes constraints and the references for each table
- **"Algorithm R8.1"** Data quality verification between the CLE contains and transformed data in the target RDBMS

Fig. 1. The general processes under the “Re-DTM”

VI. USABILITY TESTING

The impact of the developed study on information system performance was demonstrated through usability testing in the education sector. In this study, we developed a questionnaire based on its usefulness and flexibility. A (10-question) questionnaire was developed, and 35 people responded. Descriptive statistics were used to describe usability test results. The following figures show the case study’s overall results for each usability aspect.

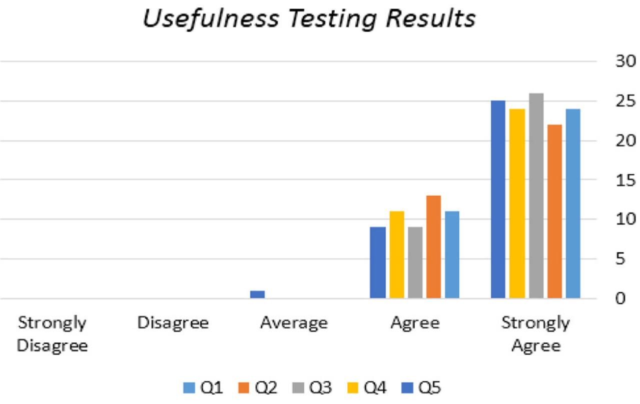


Fig. 2. The first usability test

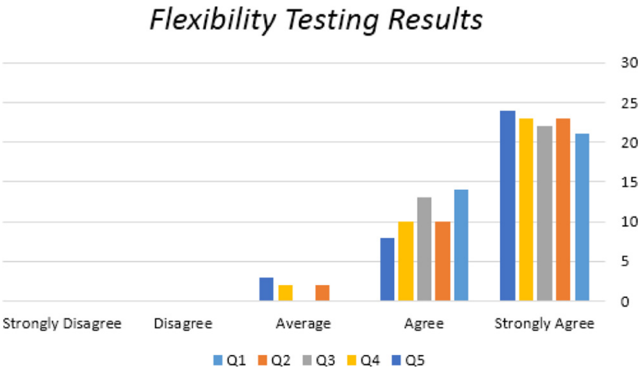


Fig. 3. The second usability test

VII. CONCLUSIONS

This study presents a new method for data transformation between relational databases. The system can connect numerous RDBMS sources and simultaneously transform the desired data into multiple target databases, indicating multiple sources with various relational database targets. Furthermore, Re-DTM automatically generates the LgS of the transformed RDBMS on the target side. In the target part, the user is not required to design or create the database structure. Based on the developed method, all database objects can be edited and handled on both the source and target sides. Experiments show that the developed method can increase the performance of information systems by offering a uniform method for accessing numerous relational databases and transforming the data with its LgS to any other type of RDBMS. In addition, decision support systems can act on a variety of inputs, including relational databases. Finally, in the future, a methodology will be developed to provide data transformation between various systems. The developed study must be expanded to support NoSQL databases, which improves the ability of information systems to interact with relational and non-relational databases.

REFERENCES

[1] S. Ristić, S. Aleksić, M. Čeliković, V. Dimitrieski, and I. Luković, “Database reverse engineering based on meta-models,” *Open Computer Science*, vol. 4, no. 3, pp. 150-159, Oct. 2014. DOI: 10.2478/s13537-014-0218-1.

[2] Y. Cheng, P. Ding, T. Wang, W. Lu, and X. Du, “Which category is

better: Benchmarking relational and graph database management systems," *Data Science and Engineering*, vol. 4, no. 4, pp. 309-322, Nov. 2019. DOI: 10.1007/s41019-019-00110-3.

[3] W. Lu, J. Hou, Y. Yan, M. Zhang, X. Du, and T. Moscibroda, "MSQL: Efficient similarity search in metric spaces using SQL," *The VLDB Journal*, vol. 26, no. 6, pp. 829-854, Dec. 2017. DOI: 10.1007/s00778-017-0481-6.

[4] H. Won, M. C. Nguyen, M. -S. Gil, Y. -S. Moon, and K. -Y. Whang, "Moving metadata from ad hoc files to database tables for robust, highly available, and scalable HDFS," *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2657-2681, Mar. 2017. DOI: 10.1007/s11227-016-1949-7.

[5] A. Prabhune, R. Stotzka, V. Sakharkar, J. Hesser, and M. Gertz, "MetaStore: An adaptive metadata management framework for heterogeneous metadata models," *Distributed and Parallel Databases*, vol. 36, no. 1, pp. 153-194, Oct. 2017. DOI: 10.1007/s10619-017-7210-4.

[6] J. Oh, W. H. Ahn, and T. Kim, "Automatic extraction of dependencies between web components and database resources in java web applications," *Journal of Information and Communication Convergence Engineering*, vol. 17, no. 2, pp. 149-160, Jun. 2019. DOI: 10.6109/jicce.2019.17.2.149.

[7] B. Walek and C. Klimes, "A methodology for data migration between different database management systems," *International Journal of Computer and Information Engineering*, vol. 6, no. 5, pp. 536-541, May. 2012. DOI: 10.5281/zenodo.1330271.

[8] P. Martins, F. Sá, C. Wanzeller, and M. Abbasi, "A performance study on different data load methods in relational databases," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, Portugal, pp. 1-7, 2019. DOI: 10.23919/CISTI.2019.8760615.

[9] P. Atzeni, L. Bellomarini, and F. Bugiotti, "EXLEngine: Executable schema mappings for statistical data processing," in *Proceedings of the 16th International Conference on Extending Database Technology*, Genoa, Italy, pp. 672-682, 2013. DOI: 10.1145/2452376.2452455.

[10] S. -C. Haw, E. Soong, N. A. Amirah, and A. Amin, "XMapDB-Sim: Performance evaluation on model-based XML to relational database mapping choices," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 7, no. 2, pp. 551-566, Aug. 2017. DOI: 10.11591/ijeecs.v7.i2.pp551-566.

[11] G. V. Machado, Í. Cunha, A. C. Pereira, and L. B. Oliveira, "DOD-ETL: Distributed on-demand ETL for near real-time business intelligence," *Journal of Internet Services and Applications*, vol. 10, no. 1, pp. 1-15, Nov. 2019. DOI: 10.1186/s13174-019-0121-z.

[12] A. Nabli, S. Bouaziz, R. Yangui, and F. Gargouri, "Two-ETL phases for data warehouse creation: Design and implementation," in *East European Conference on Advances in Databases and Information Systems*, Poitiers, France, pp. 138-150, 2015. DOI: 10.1007/978-3-319-23135-8_10.

[13] P. Kathiravelu, A. Sharma, H. Galhardas, P. V. Roy, and L. Veiga, "On-demand big data integration," *Distributed and Parallel Databases*, vol. 37, no. 2, pp. 273-295, Sep. 2019. DOI: 10.1007/s10619-018-7248-y.

[14] G. W. Sasmito, D. S. Wibowo, and D. Dairoh, "Implementation of rapid application development method in the development of geographic information systems of industrial centers," *Journal of Information and Communication Convergence Engineering*, vol. 18, no. 3, pp. 194-200, Sep. 2020. DOI: 10.6109/jicce.2020.18.3.194.

[15] W. C. Alisawi, A. A. A. Hussain, and W. A. Alawsi, "Estimate model of system management for database security," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1391-1394, Jun. 2019. DOI: 10.11591/ijeecs.v14.i3.pp1391-1394.

[16] N. A. Emran, "Data completeness measures," in *Proceedings of Pattern Analysis, Intelligent Security and the Internet of Things*, Malacca, Malaysia, pp. 117-130, 2015. DOI: 10.1007/978-3-319-17398-6_11.

[17] J. Ji and Y. Chung, "k-NN join based on LSH in big data environment," *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 2, pp. 99-105, Jun. 2018. DOI: 10.6109/jicce.2018.16.2.99.

[18] V. Theodorou, A. Abelló, W. Lehner, and M. Thiele, "Quality measures for ETL processes," in *International Conference on Data Warehousing and Knowledge Discovery*, Munich, Gemany, pp. 9-22, 2014. DOI: 10.1007/978-3-319-10160-6_2.

[19] D. P. Ballou and H. L. Pazer, "Modeling completeness versus consistency tradeoffs in information decision contexts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 1, pp. 240-243, Jan.-Feb. 2003. DOI: 10.1109/TKDE.2003.1161595.

[20] N. M. Muddasir and K. Raghuveer, "Study of methods to achieve near real time ETL," in *2017 International Conference on Current Trends in Computer, Electrical, Electronics, and Communication* (CTCEEC), Mysore, India, pp. 436-441, 2017. DOI: 10.1109/CTCEEC.2017.8455002.

[21] A. Prema and A. Pethalakshmi, "Novel approach in ETL," in *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, Salem, India, pp. 429-434, 2013. DOI: 10.1109/ICPRIME.2013.6496515.

[22] P. Tiwari, "Advanced ETL (AETL) by integration of PERL and scripting method," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, vol. 3, pp. 1-5, 2016. DOI: 10.1109/INVENTIVE.2016.7830102.

[23] M. Radonić and I. Mekterović, "ETLator-a scripting ETL framework," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, pp. 1349-1354, 2017. DOI: 10.23919/MIPRO.2017.7973632.

[24] N. E. Moukhi, I. El Azami, and A. Mouloudi, "X-ETL: A new method for designing multidimensional models," in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco, pp. 1-6, 2017. DOI: 10.1109/CloudTech.2017.8284704.

[25] B. Pan, G. Zhang, and X. Qin, "Design and realization of an ETL method in business intelligence project," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Chengdu, China, pp. 275-279, 2018. DOI: 10.1109/ICCCBDA.2018.8386526.

[26] M. A. Maatuk, A. Ali, and N. Rossiter, "Semantic enrichment: The first phase of relational database migration," in *Innovations and Advances in Computer Sciences and Engineering*, pp. 373-378, Dec. 2009. DOI: 10.1007/978-90-481-3658-2_65.

[27] L. Stanescu, M. Brezovan, and D. D. Burdescu, "Automatic mapping of MySQL databases to NoSQL MongoDB," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Gdansk, Poland, pp. 837-840, 2016.

[28] A. Ciobanu, hr-schema-mysql. 2021. Accessed: Oct. 15, 2021. [Online]. Available: https://github.com/nomemory/hr-schema-mysql/blob/0c3c8f322e607c5249de8adb8e43c0c08351d47c/hr-schema-mysql.sql.

[29] S. H. Adi, "Introduction to spatial and tabular data analysis with R," *Cover Dalam*, pp. 42, Nov. 2019.

Forat Falih Hasan
He was born in Kirkuk, Iraq, in 1986. He received the BSc.D in Manage. Information Systems in 2010, Masters Degree in Information Technology from IEC College Of Engineering & Technology/Mahamaya Technical University-India in 2012, and is pursuing Ph.D in Information Technology from the School of Computing, Universiti Utara Malaysia (UUM). His research interests include information systems, management information systems, database systems, big data, data warehouses, IoT, data quality, and business intelligence.

Muhamad Shahbani Abu Bakar
Received his Ph.D in Computer Science (Software Engineering), MSc (Information Technology), and BSc. Computer Science in 2009, 1999, and 2009, respectively. Currently, he is an Associate Professor in School of Computing, Universiti Utara Malaysia. After working as an analyst programmer and system analyst (1990-2000) in private and government sectors and a senior lecturer (2000-2017), he has served as Director of University Teaching and Learning, Universiti Utara Malaysia since 2018. His research interest includes software engineering, big data, cloud computing, learning analytic, educational technology, data warehouse, and business intelligence.

https://doi.org/10.56977/jicce.2022.20.4.265