Systematic literature review on open-source data warehouse tools and design trends

Jurgen Grotentraast
Student Data Science & technology
University of Twente
j.grotentraast@student.utwente.nl

Abstract

As data is still becoming a bigger asset every day, data warehousing is a very common practice for businesses to show trends, averages, and bottlenecks in processes. To create such a data warehouse, a plethora of tools are available and even more design methodologies have been proposed over the years. Current research has not given a clear overview of available open-source tools that can be used to create a data warehouse. On the other hand, research available on design and implementation trends for data warehouses only reflects everything before 2017. Therefore, the contribution of this paper is twofold. First, an overview of available open-source data warehouse tools was created based on literature and search engine results. To ensure these tools are a viable, future-proof option they should be last updated in or after 2023. Second, a systematic literature review was performed to find current trends or approaches for the research, design, development, implementation, or improvement of a data warehouse. This covers trends that have emerged since 2017 to see whether old trends are still being pursued or if new trends have emerged to continue where previous research on this topic has stopped.

1 Introduction

With the still-growing value of data in today's world, many organizations have invested in the development of a data warehouse (DW). A data warehouse is used to store data differently to efficiently analyze business data [50]. Data warehouses can be used for analyzing and improving business processes [118], but also to get a better understanding of for example the financial situation of an organization [72]. A data warehouse utilizes historical data to show trends, averages, and bottlenecks in a process or production chain and to show what areas of this process or production chain can be improved [29].

A data warehouse captures data from one or multiple sources, transforms the data in such a way that aggregations on this data are easy and fast to execute, and finally loads this data into the data warehouse database. This process is called extract-transform-load (ETL). Over the years many tools and software solutions have been developed to aid people in this process. Some tools are purely programming libraries or extensions that help the user to achieve what they want [20, 58, 127], whereas other software applications are developed further such that they can be used to build ETL pipelines with minimal coding. Companies like Amazon, Microsoft, and Google have developed such software applications for creating data warehouse solutions. However, these are often very expensive and require a subscription to their entire cloud platform to use them. Fortunately, over the last couple of years, open-source data warehouse tools have been developed further and further [84]. This means that open-source tools now have the same functionality as expensive enterprise solutions. Furthermore, these open-source tools allow the user to build upon the tool themselves if something is missing. For example, if a connection to a specific source of data is not yet part of the tool, the user can build a custom connector through for example an API and still extract all the data they want.

Designing a data warehouse can be done in various ways. Each approach has its advantages and disadvantages. The approach that works best for a company or person depends on various factors including the use-case of the data warehouse and the background of the designer. Over the years these approaches have been further evolved and new trends have emerged. As discussed in more detail further on, literature of the past years has extensively researched the trends of the approach of the design, development, implementation, and

improvement of a data warehouse [28, 31, 47, 70].

2 Problem statement

With the large amount of open-source tools available and the continuously changing trends in the approach or methodology of the design, development, implementation, or improvement of data warehouses, a clear overview of currently available tools and current trends is missing. Literature up until now has shown trends that were emerging up until 2018/2019 [28, 31, 47, 70], but whether those trends are still relevant and what other trends have emerged since then is unknown at the time of writing.

2.1 Research questions

Therefore, for this review, two main research questions were created. Answering these questions will give insight into current developments in open-source ETL tools as well as DW trends that have emerged over the past five years.

- **RQ1**: What open-source tools were last updated in or after 2023 are currently available?
- **RQ2**: What are trends and approaches in the research, design, development, implementation, and improvement of a data warehouse from 2018 up until 2024?

3 Methodology

A literature review was conducted for both research questions, where the tools found for **RQ1** found in the literature were extended with tools found on the internet. The methodology is based on the guidelines of Kitchenham et al. [67]. Both literature studies had a population of published studies from 2018 up to and including February 2024. All papers are written in English and are published in the field of computer science. For both literature studies, Scopus was used as the library of choice as it was indicated as the most comprehensive and user-friendly literature database [53, 83].

3.1 Search strategy

For both literature studies, a search query was created through a process of trial and error to see which combination of keywords and query composition returned the best results. Next, inclusion and exclusion criteria were created. For **RQ1**, criteria were created for both the research papers as well as for the tools. For **RQ2**, criteria were only created for the papers.

3.1.1 Tools

To answer the first research question, a combination of results from literature and the internet was used. Since the goal of this research question is to find all available tools that are currently popular, using only literature was not an option as this would yield a very limited result and would not be a representative overview of the software that is available. The following search query was used for finding literature:

(open-source OR "open source") AND ("data warehouse" OR etl) AND (solution OR tool)

The search was done in the title, abstract, and keywords of the study. The result was a set of 65 papers. These papers, however, were not all relevant. Therefore several inclusion criteria (IC) for the papers were set up. As mentioned before, the goal is to find tooling which should be considered in the proposed research. Therefore, there are only two inclusion criteria.

- **IC1**: The subject of the paper should be data warehousing or ETL.
- **IC2**: The paper should mention the tools that were used for the research.

The above-mentioned IC were applied while reading the title and abstract of each paper, this resulted in 18 papers being left. These 18 papers were carefully read to find any tool that was mentioned. This initial result was extended with tools found through a Google search. For this search the following queries were used:

- open source data warehouse tools
- open source etl tools

The resulting pages included rankings of the so-called "best" ETL or data warehouse tool to forums discussing different possibilities for tools that can be used to develop and run ETL pipelines. The complete list of tools that were found can be seen in table 1. However, not all of these tools should be taken into consideration for the proposed research.

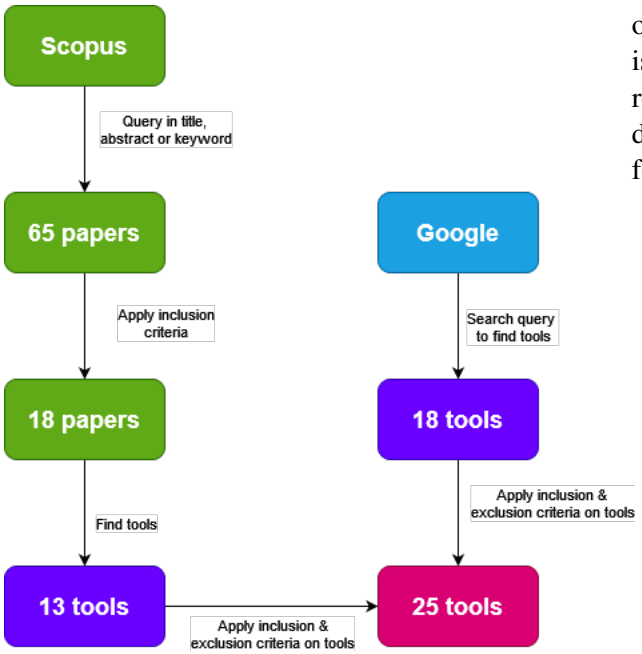


Figure 1: Tool selection process

Therefore, new inclusion and exclusion criteria for the tools were formed.

- **IC1**: The tool has to be open-source
- **IC2**: The source code should be accessible
- **IC3**: The latest release should be in or after 2023
- **EC1**: The tool is operating system specific

After applying the inclusion and exclusion criteria for tools to the initial list of tools there remained 25 tools. For a visual overview of this process see figure 1.

3.1.2 Trends & approaches

To answer the second research question, a second literature study was performed. The population was the same as for the first, except that this study was additionally limited to conference papers, articles, and book chapters. For this second part, the following search query was used:

"data warehouse" AND (design OR concept OR methodology)

The search was again done in the title, abstract, and keywords of the study. The result was a set of 743 papers. These papers, however, were not all relevant. Therefore several inclusion criteria for the papers were set up. For this study, the quality of the papers did matter, as the goal of this part is to find current trends and problems found in research in the domain of, the design of, and the development of data warehouses. Therefore, the following inclusion criteria were formed:

- **IC1**: The paper directly addresses a trend or problem with a solution or approach for the design, development, implementation, or improvement of a data warehouse
- **IC2**: The paper is peer-reviewed
- **IC3**: The paper is written in clear English
- **IC3**: The paper is available for download

An exclusion criterion was also formed to ensure the inclusion of only relevant information further.

- **EC1**: The paper discusses the implementation of a data warehouse in a specific field without explicitly addressing and explaining a trend, problem, or approach in the design, development, implementation, or improvement of a data warehouse

After applying the above criteria while reading the title and abstract there were 231 papers left initially. Some papers were initially included but were rather ambiguous. These ambiguous cases were analyzed in further detail to ensure that all non-relevant papers were excluded, this further analysis resulted in 126 papers. These ambiguous cases included terms like "detailed description of design method" in the abstract, however, in the paper itself this only included what the star schema and ETL process looked like. Since this was not relevant to this literature study, these papers were still excluded. For a visual overview of this process see figure 2.

4 Results & Discussion

The following sections show the tools that were identified in the literature and the ones that were found on Google as well as the trends and approaches that are currently of interest in research, design, and development regarding data warehouses.

Figure 2: Paper selection process

4.1 Tools

The final list of tools to answer **RQ1** consists of 25 tools. The first IC resulted in Talend, StreamSets, and Keboola being dropped. While Talend has been one of the biggest names in the open-source data warehouse industry, its software is no longer open-source as of 31st of January 2024. Keboola and StreamSets were also dropped because only a part of their tool was open-source but required the non-open-source part to work.

The second IC resulted in the exclusion of Hevo, as the source code for their platform was not accessible at all nor could the open-source license the software falls under be found. The third criterion was created to ensure that the software is still being kept up to date in terms of security and modern technologies. Therefore, Scriptella was excluded for example. Scriptella's latest release was in October 2019 and no new release has been announced or planned since then. The software also uses rather outdated technology and is therefore not seen as a future-proof solution.

The exclusion criterion resulted in Open XDMoD being excluded. Open XDMoD is an ETL tool that can only run on Linux-based systems. This is not necessarily a bad thing, however, in this review, the tools should depend on areas like use cases and employee knowledge not on the operating system someone is running.

The tools that were no longer included after applying the inclusion and exclusion criteria are marked in red in table 1. The left side shows the tools that were found on the internet. The right side shows tools that were found in the literature with the corresponding paper referenced.

The papers of Yu et al. and Spengler et al. [121, 140] are not mentioned in the table as these papers described the process of creating their own ETL tool from scratch. While the paper of Fissore et al. is mentioned in the table, it should be noted that this paper did not use any specific tool or library. Instead, they used basic Python functionalities to take care of their ETL process. However, since these have very specific purposes these are not taken into consideration. A more comprehensive overview of the included tools can be found in appendix A. This appendix shows some basic information about each tool including if the tool uses a specific programming language or is more low/no code; the size of the GitHub contributors and the number of stars the repository has; if the tool has a non-open-source paid option with more functionalities; and an optional small note for interesting capabilities, weaknesses or other noteworthy findings of the tool. This information was gathered through the website of each respective tool.

As mentioned in the methodology in section 3.1.1, the results consist of tools found in literature and through a search engine. The reason for this is that most tools that are used daily are used by companies that are not involved in research. Furthermore, the tools themselves are not created for research purposes but instead are developed as business tools to aid businesses in gaining business intelligence and performing data analytics. Therefore, these tools can not be found in literature studies. To ensure this, tools that were found through the search engine were also queried through Google Scholar to see if the initial literature search might have missed papers in which these tools were mentioned. These extra searches did not yield any new results.

Tools found on web	Tools found in literature
Airbyte	Apache Druid [38]
Apache Airflow	Apache Hadoop [38, 120, 139]
Apache Beam	Apache Hive [25, 38, 55, 139]
Apache Camel	Apache Kafka [38]
Apache Hop	Apache Spark [120]
Apache NiFi	Hevo Data [123]
Apache SeaTunnel	OpenXDMoD [33]
CloudQuery	Pentaho Community Edition [41, 123, 145]
Dagster	Python libraries* [43]
DBT	R_etl [19, 20]
Keboola	Scriptella [19, 20]
Kestra	StreamSets [123]
Knime Analytics Platform	Talend [40, 123]
Mage	
Meltano	
Prefect	
PipelineWise	
Singer	

Table 1: The complete list of tools that were found before applying the criteria. The tools that were excluded after applying the criteria are marked in red. Tools on the right were found in literature, and tools on the left were found through the accommodating web search.
*The Python libraries include: Ethereum-etl [26], Luigi, Petl [19, 20], and Pygrametl[19, 20, 58, 127]

4.2 Trends in literature

The following sections will discuss the trends found in the literature regarding data warehouse design in more detail and therefore answer **RQ2**. The papers were divided into the following six categories:

1. Data warehouse architecture
2. Data warehouse design
3. Data types
4. ETL
5. Performance
6. Schema design

Figure 3 shows the distribution of the published papers over the years for each category. In each section one of the categories is further divided into trends regarding specific topics, these are presented in tabular form and explained in more detail afterwards.

4.2.1 Data Warehouse architecture

The biggest trend that emerged in the literature on data warehouse architecture is the development of the Data Lake (DL). No papers were found that were published in 2018, but in 2019 several papers were published describing what a DL is; how it works; why it might be better than a traditional DW combined with limitations of traditional DWs; but also challenges that might need to be overcome to successfully use a DL [45, 59, 105]. This trend continued through to 2023 with the introduction of new design approaches; more DW limitations compared to the use of a DL; DL implementations with findings on challenges that were overcome; and finally in 2023 a paper describing the DL so far [30, 74, 76, 88, 116, 119]. These publications show a trend of high-level research on DL still focussed on what it is and what its benefits are. The challenges mentioned in the papers do change indicating that the DL is evolving and getting better although this is not specifically reflected in the topics discussed in the literature on DLs. A reason for this could be that this development takes place more on the commercial side rather than the research side and is therefore only reflected in research afterward.

Other topics that were mentioned in combination with DL throughout time were data quality and metadata systems. Both topics started emerging in 2019 [35, 113, 115], but continued throughout time with new studies on data quality published in 2022

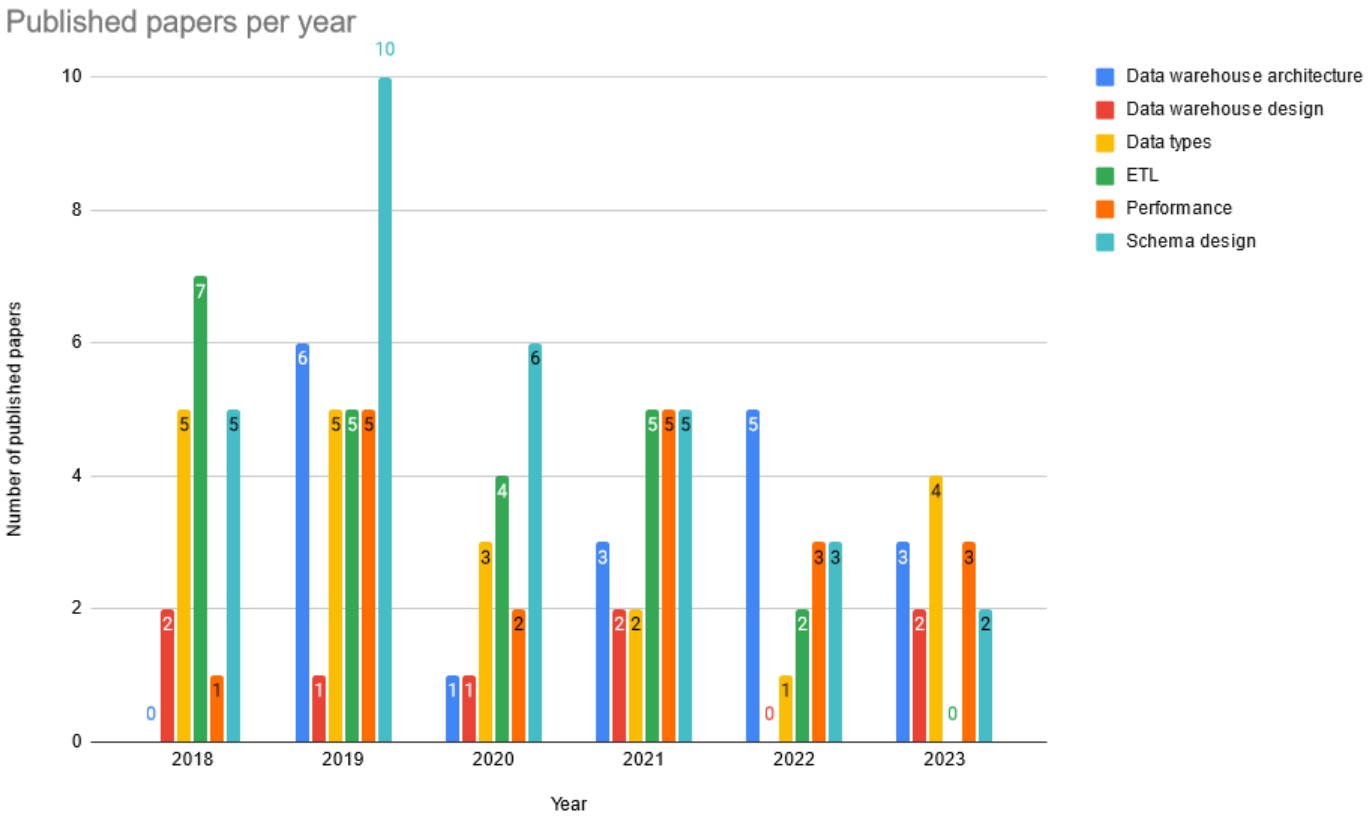


Figure 3: The number of papers published each year for every category

[10] and studies on metadata published in 2021 [114]. Although not a definitive trend yet, it shows the divergence of high-level research previously discussed into more specific problems and areas that are now tackled within the literature.

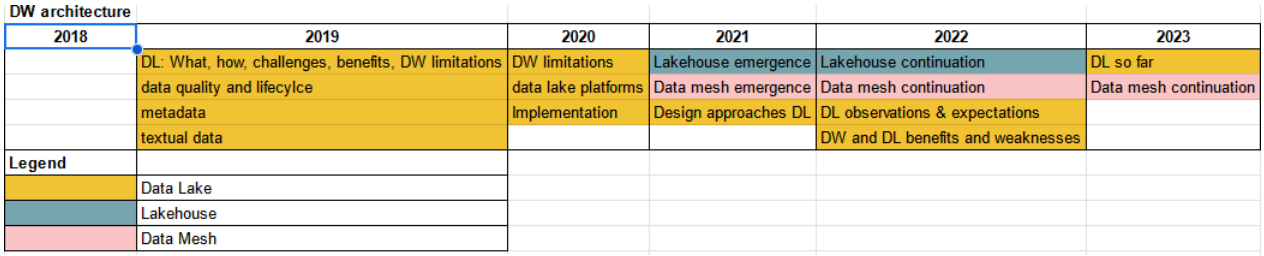
This divergence from high-level research is also seen in the other topics that were found. First is the emergence of the Lakehouse and the Data Mesh in 2021 [78, 87]. The Lakehouse was also studied in 2022 [15] by creating a faster query engine, indicating this topic might be going much faster than the DL itself, as improvements are already introduced rather than high-level conceptualizations, even though it stemmed from the DL. The data mesh was further studied in 2023 [131], this study still focuses on the concept of a data mesh itself similar to the DL. See figure 4 for a graphical overview.

4.2.2 Data warehouse design

Even though the design method of Kimball was already introduced in 2002 [66], throughout the studied years this design method was still being used as the basis for data warehouse design. While most papers simply describe and apply the method of Kimball [54, 96, 98, 132], some studies did build upon the existing methodology and improve it by introducing several hybrid approaches [124]. This suggests that although already quite old, the methodology of Kimball is still proven useful and relevant.

In 2018, Uvidia et al also proposed a new version of another methodology for data warehouse design called HEFESTO 2.0 [130] however this did not catch on as no other papers on this topic were published between 2018 and 2024. Furthermore, no new design methods were developed during the studied years nor were other design methods covered or improved.

In 2023 there were two more papers published regarding the topic of data warehouse design. One covered the implications of Brewer's rule in data warehouse design [100] and the other discussed an approach and several guidelines for creating a sustainable green data warehouse [73]. Whether these topics become relevant trends in data warehouse design is yet to be seen as they were published near the end of the covered time period.



DW architecture					
2018	2019	2020	2021	2022	2023
	DL: What, how, challenges, benefits, DW limitations	DW limitations	Lakehouse emergence	Lakehouse continuation	DL so far
	data quality and lifecylce	data lake platforms	Data mesh emergence	Data mesh continuation	Data mesh continuation
	metadata	Implementation	Design approaches DL	DL observations & expectations	
	textual data			DW and DL benefits and weaknesses	
Legend					
	Data Lake				
	Lakehouse				
	Data Mesh				

Figure 4: Categorizations of trends for DW architecture

The one trend that was found only confirms that the method developed by Kimball et al. in 2002 remains relevant to this day. See figure 5 for a graphical overview.

4.2.3 Data types

In 2018, a few trends arose regarding specific data types. First, two papers started researching ways to model trajectory data [9, 141]. Second, studies began working on incorporating Linked Open Data (LOD) and semantic data into the ETL process [17, 63, 64]. In 2019 LOD continued to be a topic of interest [65], while documented-oriented databases and NoSQL became a topic too either by itself [6, 21] or in combination with geospatial data [42]. Lastly, for 2019, interest was shown in Internet of Things (IoT) data [95]. In 2020 LOD was still being researched [18] and IoT data was also still being researched [94]. 2020 also showed the first sign of mixing two trends, with one paper studying the combination of trajectory and semantic data to see if trajectory data could be fit into a semantic data structure [107]. In 2021 the NoSQL data structure came around again with two more papers [13, 22] and in 2022 Khalil et al. looked deeper into one type of NoSQL database, namely a graph-oriented one [62].

Several of the papers mentioned in the paragraph above that look into NoSQL [6, 13, 21, 22, 62, 75] are closely related to the Data Lake and Lakehouse architectures that were discussed in section 4.2.1. These Data Lake and Lakehouse architectures are needed to store the NoSQL or document-oriented data that is discussed in this section. Furthermore, research into semantic data or LOD is also very popular as will also be seen in section 4.2.4 and section 4.2.6. The other trends on IoT and trajectory data were smaller but did show that specific data types or the nature of the gathered data does impact the ETL or Data Warehouse design. See figure 6 for a graphical overview.

4.2.4 ETL

In 2018 three main trends emerged regarding ETL design. The first, while closely related and slightly overlapping with section 4.2.3, is research on ETL design based on the type of data [16, 80, 126]. The second trend is regarding ensuring or enhancing data quality within the ETL process [89]. The third trend is regarding real-time ETL, with one paper focussing on real-time ETL as a whole [82], one paper focussing on using semantic data for real-time ETL [17], thus combining the first and third trend, and one paper more specifically on utilizing Change Data Capture (CDC) to achieve real-time ETL [27].

2019 mostly continued the second trend, with two more papers on assuring data quality [5, 122] and one paper on data enrichment using data mining [79]. Furthermore, two papers were published on expressing ETL processes using Business Process Modeling Notation (BPMN) to make ETL processes more interchangeable across tools [7, 8]. However, this topic did not become more relevant throughout the years.

In 2020 the second trend on data quality was again continued with one paper developing a framework for more general use to ensure a minimum level of data quality [4]. Furthermore, the third trend on real-time ETL continues by mixing with the second trend on data quality with one paper. This paper studies the anomalies that arise when dealing with real-time ETL and how to overcome them [81]. Then a small trend emerges with ETL design based on metadata [134]. Finally, there is one paper published on quality metrics for the entire ETL process [108], however, this does not evolve into an actual trend.

DW design					
2018	2019	2020	2021	2022	2023
Kimball	Kimball	Kimball	Kimball/hybrid		Green DW
HEFESTO					Brewer's rule
Legend					
	Kimball				
White	No trend				

Figure 5: Categorizations of trends for DW design

Data types					
2018	2019	2020	2021	2022	2023
Integration of trajectory data	Document-oriented database/NOSQL	Semantic trajectory *	NoSQL	Graph-oriented NoSQL	Trajectory DW
Trajectory ETL with graph	Geospatial in document-oriented db	LOD			Hybrid NoSQL
LOD/semantic data	LOD	IOT			NoSQL
	IOT				IOT
Legend					
	Trajectory data				
	LOD & semantic				
	NoSQL				
	IoT data				
	Semantic trajectory				

Figure 6: Categorizations of trends for Data types

This small trend on metadata has one more paper in 2021 [90], however, if this trend continues beyond 2021 is not entirely certain. No other papers on this topic were found in the population, but it might become more relevant in the future. Furthermore, in 2021 two more papers on ETL design based on data type were found. One on the impact of user-generated content on ETL design [133] and one on high-level ETL for semantic data warehouses [34]. Again there is a correlation between these papers and the papers covered in section 4.2.3, but [34, 133] focus more on the ETL whereas the papers in 4.2.3 focus more on the entire Data Warehouse. Lastly, for 2021, there are two more papers on real-time ETL [32, 44].

In 2022 only the trend on the topic of data quality and enrichment remained with two more papers, one on data cleaning inside the ETL process [137] and one on dynamic ETL for handling missing data [12]. No papers were found on any of these topics in 2023.

Overall, these results mostly indicate the importance of data quality. Without clean data, a DW will not be of any help. The trends on types presented in this section and section 4.2.3 show that the type of data or data source will also greatly impact both the ETL process and the DW design. Lastly, many papers on real-time ETL were published showing the importance of the availability of the most up-to-date data. See figure 7 for a graphical overview.

4.2.5 Performance improvement

The biggest trend found regarding performance is regarding data placement and partitioning strategies. This trend started in 2018 with one paper [1] with a strategy for predictable query times. In 2019 four more studies were published proposing new strategies to improve query time [37, 61, 103, 104]. One more study was published on this topic in 2020 proposing the decoupling of data management and computation [77]. In 2021 one study was published shifting the focus onto data placement strategies for Big data by dividing the data into themes [85] and one other study closely related to the topic of data placement which focused on materialized views to enhance query time [97]. In 2022 three more papers were published, the first paper focused on using data mining to find the best partitioning strategy [101]. The second paper focuses on a graph-oriented framework for analytical processing [62]. The third paper proposed another novel design for a

ETL					
2018	2019	2020	2021	2022	2023
Big data ETL	BPMN for ETL	Near real time ETL for big data	Metadata ETL	Dynamic ETL	
NoSQL	Quality assurance/data validation	Quality metrics of ETL	User-generated content ETL	Data cleaning	
Variety of data	Data mining	Specific ETL tool metadata based	Near real time ETL		
Ontology based ETL		Data quality	Semantic ETL		
Cleaning			Virtual DW		
Near real time ETL					
Legend					
	Data type-based ETL				
	Data quality				
	Near real time ETL				
	Metadata-based ETL				
White	No trend				

Figure 7: Categorizations of trends for ETL

distributed big data warehouse [102]. This trend seemed to end in 2022 as afterward no more papers were published however performance will always remain a relevant topic, especially with the rise in popularity of real-time ETL as discussed in section 4.2.4. This trend showed clear progress throughout the years where it started by making query time predictable, shifted quickly to improving query times for DWs in various ways, and lastly took that knowledge and applied it to Big Data problems which became more relevant over the years.

Next to this main trend, three smaller trends were forming. The first of these trends started in 2019 and focuses on the efficient joining of tables. The first paper proposes to use a GPU-based solution for the bitmap join indexes selection problem [129]. This trend continued in 2021 with one more paper using two novel joining algorithms to improve query time [11]. The second of these trends was more focused on a single solution created in 2020 called Tempura [135] which was further developed in 2023 [136]. It is yet to be seen how far this solution will be developed. The third and last of these trends is completed within the same year and therefore not quite a trend yet but still noteworthy. In 2021 two papers were published on the development of a PatchIndex structure. The first paper introduces the concept [69] and the second paper improves the results of the first to allow for efficient updates next to the existing efficient read-only queries [68].

Finally, two more papers were published not related to a specific trend. The first paper used Amdahl's and Gustafson's laws to design a decentralized clustered DW to improve parallel efficiency [99]. The last paper proposed a web ETL process that splits the workload for the ETL process related to web applications between the input device and the server [2].

Overall, the performance improvement was mainly focused on data placement and partitioning, where the ideas established in earlier years were later adapted to work in the domain of big data. The three smaller trends showed other methods to increase performance, but these methods did not establish the same kind of popularity. See figure 8 for a graphical overview.

4.2.6 Schema design

The biggest trend found regarding schema design is schema detection, generation, and evolution. Within this trend, most research was focused on automatic schema design and automatic schema evolution. The first paper on this topic was published in 2018 [125]. 2018 also saw the start of two other trends, the first regarding temporal Data Warehouses (DWs)[48]. A temporal DW is used to handle time-varying data in dimensions. The second trend is regarding the data cube [14]. A Data Cube design method allows for easier decision-making.

In 2019 research on schema detection and evolution continued with one more paper on schema detection from document-oriented data [21]. In the same year, research on temporal DWs continued with one paper [3]. Meanwhile, Data Cube's popularity rose with two more papers [23, 36]. Furthermore, two more trends were found

Performance					
2018	2019	2020	2021	2022	2023
Data model for predictable execution time	Dynamic data placement	Decoupling data management and computation	Data access/joining algorithm	Physical design through data mining	Cost-based optimizer
	Physical design/partitioning	Cost-based optimizer	Materialized views	Graph-oriented framework	Decentralized cluster
	GPU-based BJISP		Big data integration	Physical design general	Divided ETL
			PatchIndex		
Legend					
	Data placement & partitioning				
	Table join optimization				
	Query-plan optimizer				
White	No trend				

Figure 8: Categorizations of trends for performance

in 2019. The first trend is the Data Vault design [71]. The Data Vault method is most popular for allowing rebuilding the DW structure if the business model changes as well as dynamically expanding the DW model. The second trend is ontology-based schema design, which was also found in the section on data types, see section 4.2.3. In 2019 four studies on ontology-based schema design were published [39, 142–144].

The introduction of the ontology-based schema design was immediately used to combine automatic schema design with ontologies in 2020, combining the first trend with this recently started ontology trend with a paper on automatic schema design from an ontology [52]. Other works on schema detection and evolution were also published in 2020, with again the rise of applications in the domain of big data [86]. Related to this topic a way to generate DW schemas from OLAP queries [56]. But also a new schema design method based on what was learned in previous work using business intelligence problem-solving thinking and a descriptive language model which allows for clearer communication surrounding a DW schema [124]. The Data Vault methodology saw a continuation with one more paper combining the Data Vault design with a Metadata Vault Repository [57].

2021 again saw more focus on applications of previous research on the domain of big data, in this case regarding semi-automatic schema generation for Big Data Warehouses [112]. Furthermore, two studies on automated schema construction using semantic data were published, one using a natural language processing framework [110] and the other study, which similarly studied automated schema construction, but now looks more closely to semi-structured data [111]. Both these studies continue this mixed trend of ontology-based design with schema generation. In 2021 the Data Vault model continued to be studied, with a book chapter on the concept of the Data Vault model [46] and a comparison between a Data Vault model and a snowflake model [49]. This development shows that the Data Vault model is being taken seriously by researchers as a contender for a DW model.

In 2022 only the trend regarding schema generation continued with two papers on automatic schema generation, one using BI requirements and natural language processing [93], the other using machine learning to find the relevant analytical measures that are important for schema design [138]. The trend on temporal DWs also continued with one paper proposing a temporal data warehouse to solve the slowly changing dimension problem [92]. 2023 showed progress in the area of the Data Cube with one paper proposing a new Hyper Lattice structure [91] and one more paper combining the data cube and a graph-based model with a temporal DW to create a temporal graph cube model [136].

Again several papers did not belong to a specific trend. First, in 2018, a study proposed a DW design methodology that involves the end users more in the design process [109]. Also in 2018, a study was published proposing a new schema for balanced performance and security in a cloud-based DW [60]. In 2019 one study not belonging to a trend was published on a data store with a dynamic structure [6]. This study does relate somewhat to the evolving data and shift to NoSQL mentioned in previous sections but does not belong to any of the identified categories in this section. In 2020, one study presented a comparison of different schema designs in terms of performance [106]. Since the paper's main topic was schema design it was placed in this category but it does relate to performance trends a bit as well which were discussed in section 4.2.5.

The biggest trend regarding schema design was the rise in automatic schema detection, generation, and evolution. Furthermore, this category showed many overlapping trends with the rise of ontology-based design, which was also seen in other sections, and the mix of the data cube with a temporal DW and a graph-based model. This last combination also relates to section 4.2.3 where graph-based solutions were also used to handle NoSQL data which in turn relates to the rise of Data Lakes and Lakehouses discussed in section 4.2.1. See figure 9 for a graphical overview.

5 Future research & approach

The information that is gathered with this review is a perfect stepping stone for future research. With the tooling, trends, and approaches that were found in this review, a framework can be created that can be used to determine which tools are most suitable for the data warehousing needs of a business or person. Before this framework can begin taking shape, the gathered information needs to be structured. The different tools should be analyzed in more depth to determine their technological and business strengths such that this can be matched with the needs of a user and the approach they want to take. Furthermore, the tools should be tested to see if they can handle certain trends that have emerged in the design and development of DWs to ensure the tools serve as a future-proof solution. For creating the framework it must also be determined how to shape the framework. For example, the framework can be a series of questions that narrow down the tools, but can also be a clear overview of the categorization from which users can see which tool would fit their needs. After this is determined the actual framework can be created.

The framework should then also be tested. This will be done in the form of a case study for Topicus .Finance. The focus within Topicus .Finance lies on three main sectors, namely pension and wealth [117], mortgages [51], and lending [24]. Within each of these sectors, Topicus .Finance has multiple software applications for individuals or companies. During the case study, one of these software applications for the lending side will be used as the basis for a case study. The application is called Fyndoo [128], and is used by almost all major banks and many smaller banks and other financial institutions that offer lending services in the Netherlands. Fyndoo is a software application that streamlines the lending application process for both the financial institution and the applicant. Within Fyndoo Topicus .Finance wants to give their clients more insight into their processes as well as help them with the reporting they need to do to "De Nederlandsche Bank" (DNB), which in turn reports to the European Central Bank (ECB). These problems can both be solved with the use of a data warehouse which means this case is very suitable as a test for the framework.

After the framework is applied on Topicus .Finance's situation, a data warehouse will be designed and built with the tools that were suggested by the framework. This implementation will also require insight into the client of Topicus .Finance. Therefore, interviews will be conducted with these clients to gather information on the processes these clients wish to monitor and which reporting needs they still have. From these interviews, the Key Performance Indicators (KPIs) can be found which will help in the design of the data warehouse. Lastly, the implemented solution has to be evaluated. First with the clients to see if the new features of Fyndoo are helpful. Second the framework has to be evaluated with Topicus .Finance to see if the implemented solution is also a good fit for them. The framework should also be further evaluated with different parties in terms of usefulness and clarity. This can be done by sending out a survey to anybody with a technological background in data science, data warehousing, or who is otherwise proficient with data. With these evaluations, it should be possible to see if the framework is useful or not.

6 Conclusion & limitations

In conclusion, this paper first discussed the available open-source DW tools that are currently available. By using a combination of literature and search engine results a list of 31 tools was found. After applying inclusion and exclusion criteria, 25 tools remained. The tools vary from full-fledged user interface with drag-and-drop capabilities for users with limited programming experience to simple Python libraries that help group tasks together into pipelines.

Schema design					
2018	2019	2020	2021	2022	2023
Temporal DW	Data vault	Data vault	Data vault	Schema generation from natural language	Data cube (hyper lattice)
Automatic schema evolution	Temporal DW	Schema evolution from queries	Semi-automatic schema design	ML-based measure detection	Temporal graph cube **
Volunteer design	Ontology-based design	Schema design for big data	Ontology-based schema generation	Temporal DW	
Data cubes	Schema from document-oriented DB	Ontology-based schema generation			
Security design in the cloud	post-mining generalized association rules	Schema comparisons			
	Data cubes	Hybrid design methodology			
	Dynamic structure				
Legend					
	Temporal DW				
	Schema detection, generation & evolution				
	Data Cube model				
	Data Vault model				
	Ontolog-based design				
	Combines ontology-based with schema generation				
	Combines Data Cube with temporal data				
White	No trend				

Figure 9: Categorizations of trends for schema design

Next, the current trends and approaches in the research, design, development, implementation, or improvement of a data warehouse were investigated through a systematic literature review. Previous studies had done similar work up until 2018, this paper therefore focussed on everything from 2018 until February 2024. The final result consisted of 126 papers. For the DW architecture, the data lake and the lakehouse are becoming more popular. The concept of data lakes has been extensively researched over the past years to tackle any challenges that were faced while using a data lake. For the design approach, the work of Kimball et al. [66] is still popular. Other approaches were mentioned too, but not as often as the nine steps outlined by Kimball. For the data model design, most research was focused on automatic schema creation and automatic schema updating. Another big trend was ontology-based schema design. This ontology-based approach also emerged in research that aimed to deal with new types of data. Especially linked open data and IoT data are popular topics. Furthermore, trajectory data of moving objects was also mentioned several times. These types of data also resulted in more work on NoSQL databases as these are very suitable to deal with a plethora of data types. Research on ETL is mostly focussed on near real-time ETL as the demand for fast data analytics is growing. Another big trend in ETL research is data quality and enrichment. The importance of ensuring clean data was already clear, the current research mostly suggests new, more efficient ways of cleaning and enriching. Lastly, studies on performance improvement showed two main trends. The first is the physical design of the system in combination with data placement. The second trend is focused on query optimization and efficiency.

The limitations of this study lie in the collection of relevant literature. First of all, as mentioned before and as the results show, the literature on open-source tooling was very limited. Adding search engine results made sure a complete overview of tools could be made but the threat in this approach is that the results are not proved scientifically. The impact of this threat however is very limited, as the goal of RQ1 was to find a complete overview of software that is used on a daily basis in both professional and smaller-scaled settings.

Second of all, there is a threat of validity to the collection of relevant literature for RQ2. The threat in this part is twofold. First, data warehousing is a very hot topic for research which means that a search query like the one used in this study retrieved almost 750 papers. While the use of inclusion and exclusion criteria to narrow these papers down worked well, there is a possibility that papers were excluded even though they did contain relevant information. This threat was mitigated as much as possible by dividing the work into smaller portions to ensure that there was no lack of focus during the review process. Second, the result is currently limited to only Scopus. While it was indicated as the most comprehensive and user-friendly literature database [53, 83] and in total 126 papers were covered in this study, it is possible that other literary databases included papers that are also relevant to this topic. Nevertheless, the number of papers is seen as enough to cover the trends that have emerged since 2017.

References

[1] M. Abbasi et al. “SINGLE vs. MapReduce vs. relational: Predicting query execution time”. In: *Communications in Computer and Information Science* 928 (2018), pp. 63–74. DOI: 10.1007/978-3-319-99987-6_5. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85053858167&doi=10.1007%2f978-3-319-99987-6_5&partnerID=40&md5=ed87ddc992e9e0418a5a48fff6708562.

[2] S.Q. Abd Al-Rahman, E.H. Hasan, and A.M. Sagheer. “Design and implementation of the web (extract, transform, load) process in data warehouse application”. In: *IAES International Journal of Artificial Intelligence* 12.2 (2023), pp. 765–775. DOI: 10.11591/ijai.v12.i2.pp765-775. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143777363&doi=10.11591%2fijai.v12.i2.pp765-775&partnerID=40&md5=b48c4ca8688c4d3104fd9c3e7226725b.

[3] S. Ain El Hayat and M. Bahaj. “A temporal data warehouse conceptual modelling and its transformation into temporal object relational model”. In: *Advances in Intelligent Systems and Computing* 915 (2019), pp. 314–323. DOI: 10.1007/978-3-030-11928-7_28. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069992509&doi=10.1007%2f978-3-030-11928-7_28&partnerID=40&md5=c8ae2ea2e7c9a57f84e747821ca1c52f.

[4] T.Z. Ali et al. “A framework for improving data quality in data warehouse: A case study”. In: 2020. DOI: 10.1109/ACIT50332.2020.9300119. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85099718237&doi=10.1109%2fACIT50332.2020.9300119&partnerID=40&md5=2ef7fadda7ef02e929ce64afc61e1005.

[5] P. Amuthabala and R. Santhosh. “Robust analysis and optimization of a novel efficient quality assurance model in data warehousing”. In: *Computers and Electrical Engineering* 74 (2019), pp. 233–244. DOI: 10.1016/j.compeleceng.2019.02.003. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061199661&doi=10.1016%2fj.compeleceng.2019.02.003&partnerID=40&md5=5227a45af6ba24c2577b6dbd7bb65b2d.

[6] Y.N. Artamonov. “Building a Data Store with the Dynamic Structure”. In: *Automatic Control and Computer Sciences* 53.7 (2019), pp. 794–810. DOI: 10.3103/S0146411619070265.

[7] J. Awiti, A. Vaisman, and E. Zimányi. “From Conceptual to Logical ETL Design Using BPMN and Relational Algebra”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11708 LNCS (2019), pp. 299–309. DOI: 10.1007/978-3-030-27520-4_21. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072985913&doi=10.1007%2f978-3-030-27520-4_21&partnerID=40&md5=5b81896b66835a6e87d0d191269ecc3e.

[8] J. Awiti and E. Zimányi. “An XML Interchange Format for ETL Models”. In: *Communications in Computer and Information Science* 1064 (2019), pp. 427–439. DOI: 10.1007/978-3-030-30278-8_42. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072957643&doi=10.1007%2f978-3-030-30278-8_42&partnerID=40&md5=2ad5c4fea013823c478fead36bdf3a87.

[9] N. Azaiez and J. Akaichi. “Integrating trajectory data in the warehousing chain: A new way to handle the trajectory ELT process”. In: *Smart Innovation, Systems and Technologies* 76 (2018), pp. 353–361. DOI: 10.1007/978-3-319-59480-4_35.

[10] O. Azeroual et al. “Combining Data Lake and Data Wrangling for Ensuring Data Quality in CRIS”. In: vol. 211. C. 2022, pp. 3–16. DOI: 10.1016/j.procs.2022.10.171.

[11] O. Aziz, T. Anees, and E. Mehmood. “An Efficient Data Access Approach with Queue and Stack in Optimized Hybrid Join”. In: *IEEE Access* 9 (2021), pp. 41261–41274. DOI: 10.1109/ACCESS.2021.3064202. URL: https://www.scopus.

com/inward/record.uri?eid=2-s2.0-85102653424&doi=10.1109%2fACCESS.2021.3064202&partnerID=40&md5=d2e10d97803c087f9e6614db4c863b7a.

[12] M. Badiuzzaman Biplob and M. Mokammel Haque. "Development of an Efficient ETL Technique for Data Warehouses". In: *Lecture Notes on Data Engineering and Communications Technologies* 95 (2022), pp. 243–255. DOI: 10.1007/978-981-16-6636-0_20. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85120866259&doi=10.1007%2f978-981-16-6636-0_20&partnerID=40&md5=aaa27a3cce9c9e502ec202d0483844b1.

[13] S. Banerjee et al. "A unified conceptual model for data warehouses". In: *Annals of Emerging Technologies in Computing* 5.Special issue 5 (2021), pp. 162–169. DOI: 10.33166/AETiC.2021.05.020.

[14] D. Bantug, P. Franklin, and T. Boone. "Product Reliability and Databases: Lessons Learned". In: vol. 2018-January. 2018. DOI: 10.1109/RAM.2018.8463091. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054139495&doi=10.1109%2fRAM.2018.8463091&partnerID=40&md5=3c287dfcb28cca9094900d34c85b95fd.

[15] A. Behm et al. "Photon: A Fast Query Engine for Lakehouse Systems". In: 2022, pp. 2326–2339. DOI: 10.1145/3514221.3526054.

[16] N. Berkani, L. Bellatreche, and L. Guittet. "ETL Processes in the Era of Variety". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11310 LNCS (2018), pp. 98–129. DOI: 10.1007/978-3-662-58415-6_4. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069217253&doi=10.1007%2f978-3-662-58415-6_4&partnerID=40&md5=77e77d9a13d9b4c59712c026bebc85fd.

[17] N. Berkani, L. Bellatreche, and C. Ordonez. "ETL-aware materialized view selection in semantic data stream warehouses". In: vol. 2018-May. 2018, pp. 1–11. DOI: 10.1109/RCIS.2018.8406668. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85050878392&doi=10.1109%2fRCIS.2018.8406668&partnerID=40&md5=e2412cf55af4fee8e19e61b26b083b8d.

[18] N. Berkani et al. "The contribution of linked open data to augment a traditional data warehouse". In: *Journal of Intelligent Information Systems* 55.3 (2020), pp. 397–421. DOI: 10.1007/s10844-020-00594-w.

[19] Neepa Biswas, Anamitra Sarkar, and Kartick Chandra Mondal. "Efficient incremental loading in ETL processing for real-time data integration". In: *Innovations in Systems and Software Engineering* 16.1 (2020), pp. 53–61. DOI: 10.1007/s11334-019-00344-4.

[20] Neepa Biswas, Anamitra Sarkar, and Kartick Chandra Mondal. "Empirical Analysis of Programmable ETL Tools". In: *Communications in Computer and Information Science* 1031 (2019). Ed. by Mandal J.K. et al., pp. 267–277. DOI: 10.1007/978-981-13-8581-0_22.

[21] S. Bouaziz, A. Nabli, and F. Gargouri. "Design a data warehouse schema from document-oriented database". In: vol. 159. 2019, pp. 221–230. DOI: 10.1016/j.procs.2019.09.177.

[22] S. Bouaziz, A. Nabli, and F. Gargouri. "Towards data warehouse from open data: Case of COVID-19". In: *International Journal of Hybrid Intelligent Systems* 17.3-4 (2021), pp. 129–142. DOI: 10.3233/HIS-210010.

[23] H. Brahmi. "Post-Mining of Generalized Association Rules from Data Cubes". In: vol. 2019-January. 2019, pp. 153–158. DOI: 10.1109/ICOIN.2019.8718147. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85066746273&doi=10.1109%2fICOIN.2019.8718147&partnerID=40&md5=af5b08acf587e71ea7e425f4f46b7855.

[24] Jamie Burink. *Topicus Lending*. Ed. by Topicus Finance. URL: https://topicus.nl/en/solutions/business-lending.

[25] Jesús Camacho-Rodríguez et al. "Apache hive: From mapreduce to enterprise-grade big data warehousing". In: Association for

Computing Machinery, 2019, pp. 1773–1786. DOI: 10.1145/3299869.3314045.

[26] Giorgio Camozzi, Felix Härer, and Hans-Georg Fill. “Multidimensional Analysis of Blockchain Data Using an ETL-based Approach”. In: Association for Information Systems, 2022.

[27] H. Chandra. “Analysis of Change Data Capture Method in Heterogeneous Data Sources to Support RTDW”. In: 2018. DOI: 10.1109/ICCOINS.2018.8510574. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85057131047&doi=10.1109%2fICCOINS.2018.8510574&partnerID=40&md5=e69e1e4994120eb63f6f059b4a1c2e24.

[28] Pravin Chandra and Manoj K Gupta. “Comprehensive survey on data warehousing research”. In: *International Journal of Information Technology* 10 (2018), pp. 217–224.

[29] Surajit Chaudhuri and Umeshwar Dayal. “An overview of data warehousing and OLAP technology”. In: *SIGMOD Rec.* 26 (Mar. 1997), pp. 65–74. ISSN: 0163-5808. DOI: 10.1145/248603.248616. URL: https://doi.org/10.1145/248603.248616.

[30] Z. Chen. “Observations and Expectations on Recent Developments of Data Lakes”. In: vol. 214. C. 2022, pp. 405–411. DOI: 10.1016/j.procs.2022.11.192.

[31] Carlos Costa and Maribel Yasmina Santos. “Evaluating several design patterns and trends in big data warehousing systems”. In: *Advanced Information Systems Engineering: 30th International Conference, CAiSE 2018, Tallinn, Estonia, June 11-15, 2018, Proceedings 30*. Springer. 2018, pp. 459–473.

[32] F. De Assis Vilela and R. Rodrigues Ciferri. “A novel solution to perform real-time ETL process based on non-intrusive and reactive concepts”. In: 2021, pp. 556–561. DOI: 10.1109/CSCI54926.2021.00158. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85133909470&doi=10.1109%2fCSCI54926.2021.00158&partnerID=40&md5=8fec68a097815c1c8a2386a84dfcb381.

[33] Gregary Dean et al. “Performance Optimization of the Open XDMoD Datawarehouse”. In: Association for Computing Machinery, Inc, 2022. DOI: 10.1145/3491418.3530290.

[34] R.P. Deb Nath et al. “High-level ETL for semantic data warehouses”. In: *Semantic Web* 13.1 (2021), pp. 85–132. DOI: 10.3233/SW-210429. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85149945068&doi=10.3233%2fSW-210429&partnerID=40&md5=b6966d0abaa03f8f0d5ad73a068348e5.

[35] M. Derakhshannia et al. “Life and Death of Data in Data Lakes: Preserving Data Usability and Responsible Governance”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11938 LNCS (2019), pp. 302–309. DOI: 10.1007/978-3-030-34770-3_24.

[36] R. Djiroun, K. Boukhalfa, and Z. Alimazighi. “Designing data cubes in OLAP systems: a decision makers’ requirements-based approach”. In: *Cluster Computing* 22.3 (2019), pp. 783–803. DOI: 10.1007/s10586-018-2883-7. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85058468322&doi=10.1007%2fs10586-018-2883-7&partnerID=40&md5=33a084cc041d7d00f8ab9aeec677689d.

[37] J. Du et al. “Towards Dynamic Data Placement for Polystore Ingestion”. In: *Lecture Notes in Business Information Processing* 337 (2019), pp. 211–228. DOI: 10.1007/978-3-030-24124-7_13. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85075686723&doi=10.1007%2f978-3-030-24124-7_13&partnerID=40&md5=ea80efc564b700c48d56058dde3d2250.

[38] Paweł Dymora, Gabriel Lichacz, and Mirosław Mazurek. “Performance Analysis of a Real-Time Data Warehouse System Implementation Based on Open-Source Technologies”. In: *Lecture Notes in Networks and Systems* 737 LNNS (2023). Ed. by Zamojski W. et al., pp. 63–73. DOI: 10.1007/978-3-031-37720-4_6.

[39] E. Elamin et al. “A hybrid DW design method using a semantic resource”. In: 2019. DOI: 10.1109/ICCCEEE46830.2019.9071359. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85084278725&doi=10.1109%2fICCCEEE46830.2019.9071359&partnerID=40&md5=0ab81c3b857c7eb5e8e2d3e5d113d316.

[40] Juan Espinoza et al. “Development of an OpenMRS-OMOP ETL tool to support informatics research and collaboration in LMICs”. In: *Computer Methods and Programs in Biomedicine Update* 4 (2023). DOI: 10.1016/j.cmpbup.2023.100119.

[41] Yong-Liang Fang and Rong-Hua Ye. “Research and Implementation of ETL Algorithm Based on Kettle Cluster”. In: ed. by Pei Z. Vol. 12331. SPIE, 2022. DOI: 10.1117/12.2652244.

[42] M. Ferro, R. Lima, and R. Fidalgo. “Evaluating Redundancy and Partitioning of Geospatial Data in Document-Oriented Data Warehouses”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11708 LNCS (2019), pp. 221–235. DOI: 10.1007/978-3-030-27520-4_16.

[43] F. Fissore and F. Pirotti. “Migration of digital cartography to CityGML; a web-based tool for supporting simple etl procedures”. In: ed. by Zlatanova S., Sithole G., and Dragicevic S. Vol. 42. 4. International Society for Photogrammetry and Remote Sensing, 2018, pp. 267–274. DOI: 10.5194/isprs-archives-XLII-4-193-2018.

[44] P. Ghosh, D. Sadhu, and S. Sen. “A real-time business analysis framework using virtual data warehouse”. In: *International Arab Journal of Information Technology* 18.4 (2021), pp. 585–595. DOI: 10.34028/18/4/11. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85110497818&doi=10.34028%2f18%2f4%2f11&partnerID=40&md5=faf36c630b2b06dc0f09172e6da049fd.

[45] C. Giebler et al. “Leveraging the Data Lake: Current State and Challenges”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11708 LNCS (2019), pp. 179–188. DOI: 10.1007/978-3-030-27520-4_13.

[46] P. Gluchowski. *Data Vault as a Modeling Concept for the Data Warehouse*. 2021, pp. 277–286. DOI: 10.1007/978-3-030-84655-8_17. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85153669345&doi=10.1007%2f978-3-030-84655-8_17&partnerID=40&md5=4b76b5918995e765e92c239907d2f9b4.

[47] Matteo Golfarelli and Stefano Rizzi. “From Star Schemas to Big Data: 20 Years of Data Warehouse Research”. In: *A comprehensive guide through the Italian database research over the last 25 years* (2017), pp. 93–107.

[48] A. Gosain and K. Saroha. “Bi-temporal versioning of schema in temporal data warehouses”. In: *Advances in Intelligent Systems and Computing* 542 (2018), pp. 357–367. DOI: 10.1007/978-981-10-3223-3_34. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85021197572&doi=10.1007%2f978-981-10-3223-3_34&partnerID=40&md5=f9d2413477bcb3e5692c36ba03143291.

[49] Y. Grigoriev, E. Ermakov, and O. Ermakov. “Hadoop/Hive Data Query Performance Comparison Between Data Warehouses Designed by Data Vault and Snowflake Methodologies”. In: *Communications in Computer and Information Science* 1204 CCIS (2021), pp. 147–156. DOI: 10.1007/978-3-030-78273-3_15. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111398459&doi=10.1007%2f978-3-030-78273-3_15&partnerID=40&md5=8b29f5308c31f02fa12a828f4290c07c.

[50] Himanshu Gupta. “Selection of views to materialize in a data warehouse”. In: *Database Theory—ICDT’97: 6th International Conference Delphi, Greece, January 8–10, 1997 Proceedings 6*. Springer. 1997, pp. 98–112.

[51] Clint van Haalen. *Topicus Mortgages*. Ed. by Topicus Finance. URL: https://topicus.nl/en/solutions/mortgages.

[52] M. Hajji, M. Qbadou, and K. Mansouri. “Towards Eclipse Plug-ins for Automated Data Warehouse Design from an Ontol-

ogy". In: *Advances in Intelligent Systems and Computing* 1076 (2020), pp. 613–621. DOI: 10.1007/978-981-15-0947-6_58. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85085204271&doi=10.1007%2f978-981-15-0947-6_58&partnerID=40&md5=9cbd54b63e8bc76bc6c5f76b400a4f20.

[53] Anne-Wil Harzing and Satu Alakangas. "Google Scholar, Scopus and the Web of Science: a longitudinal and cross-disciplinary comparison". In: *Scientometrics* 106 (2016), pp. 787–804.

[54] M. Himami et al. "Utilization of Data Warehouse in Business Intelligence with Kimball Method at Company XYZ". In: 2021, pp. 146–151. DOI: 10.1109/ICAIBDA53487.2021.9689720.

[55] Voon Hou Su, Sourav Sen Gupta, and Arijit Khan. "Automating ETL and mining of ethereum blockchain network". In: Association for Computing Machinery, Inc, 2022, pp. 1581–1584. DOI: 10.1145/3488560.3502187.

[56] Z. Huo et al. "Generating multidimensional schemata from relational aggregation queries". In: *World Wide Web* 23.1 (2020), pp. 337–359. DOI: 10.1007/s11280-019-00706-9. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069200169&doi=10.1007%2fs11280-019-00706-9&partnerID=40&md5=e49bd7a8458f141f1b25451cc1538cb0.

[57] D. Jaksic, P. Poscic, and V. Jovanovic. "Conceptual model for the new generation of data warehouse system catalog". In: *Lecture Notes in Networks and Systems* 69 (2020), pp. 813–825. DOI: 10.1007/978-3-030-12388-8_55. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062898876&doi=10.1007%2f978-3-030-12388-8_55&partnerID=40&md5=4d6028059826f820a3f4894bc3f8428b.

[58] Søren Kejser Jensen et al. "pygrametl: A Powerful Programming Framework for Easy Creation and Testing of ETL Flows". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12670 LNCS (2021), pp. 45–84. DOI: 10.1007/978-3-662-63519-3_3.

[59] J. Kachaoui and A. Belangour. "Challenges and benefits of deploying big data storage solution". In: 2019. DOI: 10.1145/3314074.3314097.

[60] K. Karkouda, A. Nabli, and F. Gargouri. "CloudWar: A new schema for securing and querying data warehouse hosted in the cloud". In: 2018, pp. 6–12. DOI: 10.1109/ICCTA45985.2018.9499193. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85114879122&doi=10.1109%2fICCTA45985.2018.9499193&partnerID=40&md5=5849ae6e1541ae2f824cabebb4cadf9a.

[61] M. Kechar and S. Nait-Bahloul. "Bringing together physical design and fast querying of large data warehouses: A new data partitioning strategy". In: 2019. DOI: 10.1145/3372938.3372947. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85117541347&doi=10.1145%2f3372938.3372947&partnerID=40&md5=7be0408d0b9a2bac8a1856fd4340bb0d.

[62] A. Khalil and M. Belaissaoui. "A Graph-oriented Framework for Online Analytical Processing". In: *International Journal of Advanced Computer Science and Applications* 13.5 (2022), pp. 547–555. DOI: 10.14569/IJACSA.2022.0130564.

[63] S. Khouri and L. Bellatreche. "Consolidation of BI efforts in the LOD era for african context". In: 2018, pp. 1–10. DOI: 10.1145/3195528.3195529.

[64] S. Khouri and L. Bellatreche. "LOD for data warehouses: Managing the ecosystem co-evolution". In: *Information (Switzerland)* 9.7 (2018). DOI: 10.3390/info9070174.

[65] Selma Khouri et al. "Data cube is dead, long life to data cube in the age of web data". In: *Big Data Analytics: 7th International Conference, BDA 2019, Ahmedabad, India, December 17–20, 2019, Proceedings 7*. Springer. 2019, pp. 44–64.

[66] Ralph Kimball and Margy Ross. *The data warehouse toolkit: the complete guide to*

dimensional modeling. John Wiley & Sons, 2011.

[67] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. *Evidence-based software engineering and systematic reviews*. Vol. 4. CRC press, 2015.

[68] S. Klabe, K.-U. Sattler, and S. Baumann. "Updatable materialization of approximate constraints". In: vol. 2021-April. 2021, pp. 1991–1996. DOI: 10.1109/ICDE51399.2021.00189. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85112866100&doi=10.1109%2fICDE51399.2021.00189&partnerID=40&md5=95c93cedb7ec1df4a00342fa25d30100.

[69] S. Kläbe, K.-U. Sattler, and S. Baumann. "PatchIndex: exploiting approximate constraints in distributed databases". In: *Distributed and Parallel Databases* 39.3 (2021), pp. 833–853. DOI: 10.1007/s10619-021-07326-1. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102291632&doi=10.1007%2fs10619-021-07326-1&partnerID=40&md5=069e5d5d4c8014f875c9f93051032053.

[70] Natalija Kozmina, Laila Niedrite, and Janis Zemnickis. "Information requirements for big data projects: A review of state-of-the-art approaches". In: *Databases and Information Systems: 13th International Baltic Conference, DB&IS 2018, Trakai, Lithuania, July 1-4, 2018, Proceedings 13*. Springer. 2018, pp. 73–89.

[71] Y. Kuznetcov, M. Fomin, and A. Vinogradov. "Multidimensional information systems metadata repository development with a data warehouse structure using "data vault" methodology". In: 2019. DOI: 10.1145/3373722.3373777. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85123039944&doi=10.1145%2f3373722.3373777&partnerID=40&md5=19da256add085d9d31ef2034d55d1503.

[72] Earl Von F Lapura et al. "Development of a University Financial Data Warehouse and its Visualization Tool". In: *Procedia Computer Science* 135 (2018), pp. 587–595.

[73] Khadija Letrache, Omar El Beggar, and Mohammed Ramdani. "Green data warehouse design and exploitation". In: *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*. 2018, pp. 1–6.

[74] G. Li, W. Hu, and T. You. "Data Lake Development Status and Outlook". In: vol. 12814. 2023. DOI: 10.1117/12.3011074.

[75] G. Liu et al. "IoT Lakehouse: A New Data Management Paradigm for AIoT". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 14203 LNCS (2023), pp. 34–47. DOI: 10.1007/978-3-031-44725-9_3.

[76] R. Liu, H. Isah, and F. Zulkernine. "A Big Data Lake for Multilevel Streaming Analytics". In: 2020. DOI: 10.1109/IBDAP50342.2020.9245460.

[77] Z. Liu et al. "Towards Elastic Data Warehousing by Decoupling Data Management and Computation". In: 2020, pp. 52–57. DOI: 10.1145/3416921.3416935. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092690665&doi=10.1145%2f3416921.3416935&partnerID=40&md5=3d2e3c538999b80a18ccf1f788c74f02.

[78] I.A. Machado, C. Costa, and M.Y. Santos. "Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures". In: vol. 196. 2021, pp. 263–271. DOI: 10.1016/j.procs.2021.12.013.

[79] M. Madhikerrni and K. Främling. "Data discovery method for Extract- Transform-Load". In: 2019, pp. 174–181. DOI: 10.1109/ICMIMT.2019.8712027. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85066473340&doi=10.1109%2fICMIMT.2019.8712027&partnerID=40&md5=d6dc929d80bb97a623d347301d8e67e7.

[80] H. Mallek et al. "BigDimETL with NoSQL Database". In: vol. 126. 2018, pp. 798–807. DOI: 10.1016/j.procS.2018.08.014. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85056669248&doi=10.1016%2fj.procS.

2018.08.014&partnerID=40&md5=bee37b18cfa1fcab2a664caf6cbde65d.

[81] N. Mohammed Muddasir and K. Raghuveer. "A Novel Approach to Handle Huge Data for Refreshment Anomalies in Near Real-Time ETL Applications". In: *Advances in Intelligent Systems and Computing* 1154 (2020), pp. 545–554. DOI: 10.1007/978-981-15-4032-5_50. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088288050&doi=10.1007%2f978-981-15-4032-5_50&partnerID=40&md5=9e32ed6c71271d7f0991f6dd58030940.

[82] N. Mohammed Muddasir and K. Raghuveer. "Study of Methods to Achieve Near Real Time ETL". In: 2018, pp. 436–441. DOI: 10.1109/CTCEEC.2017.8455002. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054082609&doi=10.1109%2fCTCEEC.2017.8455002&partnerID=40&md5=0afb024055fe2e8f7de9a94a71140363.

[83] Philippe Mongeon and Adèle Paul-Hus. "The journal coverage of Web of Science and Scopus: a comparative analysis". In: *Scientometrics* 106 (2016), pp. 213–228.

[84] Salwa Mohammed Nejres. "Analysis of data warehousing and data mining in education domain". In: *International Journal of Advances in Computer Science and Technology* 4.04 (2015).

[85] W. Nie et al. "Design of big data integration platform based on hybrid hierarchy architecture". In: 2021, pp. 135–140. DOI: 10.1109/BigDataSE53435.2021.00028. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85127188577&doi=10.1109%2fBigDataSE53435.2021.00028&partnerID=40&md5=dc847819d50ed1901a42c26f9cd5cd07.

[86] M. Nogueira, J. Galvão, and M.Y. Santos. "A data modelling method for big data warehouses". In: *Lecture Notes in Business Information Processing* 381 LNBIP (2020), pp. 85–98. DOI: 10.1007/978-3-030-44322-1_7. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083978977&doi=10.1007%2f978-3-030-44322-1_7&partnerID=40&md5=c9cf6c02a94e2638a1ede783883cd6d8.

[87] D. Orescanin and T. Hlupic. "Data Lakehouse - A Novel Step in Analytics Architecture". In: 2021, pp. 1242–1246. DOI: 10.23919/MIPRO52101.2021.9597091.

[88] L. Oukhouya et al. "Designing Hybrid Storage Architectures with RDBMS and NoSQL Systems: A Survey". In: *Lecture Notes in Networks and Systems* 637 LNNS (2023), pp. 332–343. DOI: 10.1007/978-3-031-26384-2_29.

[89] B. Pan, G. Zhang, and X. Qin. "Design and realization of an ETL method in business intelligence project". In: 2018, pp. 275–279. DOI: 10.1109/ICCCBDA.2018.8386526. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85050088457&doi=10.1109%2fICCCBDA.2018.8386526&partnerID=40&md5=bda0c38d705d26a3aee6d9a95125124f.

[90] P. Panfilov and A. Suleykin. "Chapter 8 Building Resilience into the Metadata-Based ETL Process Using Open Source Big Data Technologies". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12660 LNCS (2021), pp. 139–153. DOI: 10.1007/978-3-030-70370-7_8. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102065764&doi=10.1007%2f978-3-030-70370-7_8&partnerID=40&md5=827b94a3b5dc826c98bb84c3e1590cf1.

[91] A.K. Phogat and S. Mann. "Hyper Lattice Structure for Data Cube Computation". In: *Lecture Notes in Networks and Systems* 572 (2023), pp. 697–705. DOI: 10.1007/978-981-19-7615-5_57. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85152539896&doi=10.1007%2f978-981-19-7615-5_57&partnerID=40&md5=6b6d805f62eb977250e52d607703bb39.

[92] T. Phungtua-Eng and S. Chittayasothorn. "Information Integration and Multiple Slowly Changing Dimensions Modeling". In: 2022, pp. 214–222. DOI: 10.1145/3547578.3547611. URL: https://www.scopus.com/inward/

record . uri ? eid = 2 - s2 . 0 - 85139167135&doi=10.1145%2f3547578.3547611 & partnerID = 40 & md5 = 4d93ac5013123eecfd429f53eb09532c.

[93] C.A. Pizarro, G. Novillo, and G. Montejano. "Automatic Data Warehouse Generation Model from BI Requirements in Natural Language". In: *Smart Innovation, Systems and Technologies* 259 SIST (2022), pp. 13–21. DOI: 10.1007/978-981-16-5792-4_2. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85116867788&doi=10.1007%2f978-981-16-5792-4_2&partnerID=40&md5=bb438fb2c39c685075ec19eaf6e04fa8.

[94] J.E. Plazas et al. "Self-service Business Intelligence over On-Demand IoT Data: A New Design Methodology Based on Rapid Prototyping". In: *Communications in Computer and Information Science* 1259 CCIS (2020), pp. 84–93. DOI: 10.1007/978-3-030-54623-6_8.

[95] A. Rahman, Ermatita, and D. Budianta. "Data Warehouse Design for Soil Nutrients with IoT Based Data Sources". In: 2019, pp. 181–186. DOI: 10.1109/ICIMCIS48181.2019.8985209.

[96] R. Rahutomo, R.A. Putri, and B. Pardamean. "Building Datawarehouse for Educational Institutions in 9 Steps". In: 2018, pp. 128–133. DOI: 10.1109/INAPR.2018.8627010.

[97] A.R. Raipurkar and M.B. Chandak. "Optimized execution method for queries with materialized views: Design and implementation". In: *Journal of Intelligent and Fuzzy Systems* 41.6 (2021), pp. 6191–6205. DOI: 10.3233/JIFS-202821. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85122032374&doi=10.3233%2fJIFS-202821&partnerID=40&md5=24ec0bf525ac0fe68c3f1a0979430e4a.

[98] P.P. Ramadhani, S. Hadi, and R. Rosadi. "Implementation of Data Warehouse in Making Business Intelligence Dashboard Development Using PostgreSQL Database and Kimball Lifecycle Method". In: 2021, pp. 88–92. DOI: 10.1109/ICAIBDA53487.2021.9689697.

[99] R. Raman et al. "Designing of a Decentralized Cluster Data Warehouse using Amdahl's and Gustafson's Law". In: 2023, pp. 343–348. DOI: 10.1109/I-SMAC58438.2023.10290405. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85177614027&doi=10.1109%2fI-SMAC58438.2023.10290405&partnerID=40&md5=ba681357265a40b69fb267c904cd32e4.

[100] Ramakrishnan Raman et al. "Implications of Brewer's Rule in Data Warehouse Design". In: *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE. 2023, pp. 349–354.

[101] Y. Ramdane et al. "A Data Mining Approach to Guide the Physical Design of Distributed Big Data Warehouses". In: *Studies in Computational Intelligence* 1004 (2022), pp. 107–125. DOI: 10.1007/978-3-030-90287-2_6. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85127032066&doi=10.1007%2f978-3-030-90287-2_6&partnerID=40&md5=c2b12bdfc0043abc63f266689600c6ea.

[102] Y. Ramdane et al. "Building a novel physical design of a distributed big data warehouse over a Hadoop cluster to enhance OLAP cube query performance". In: *Parallel Computing* 111 (2022). DOI: 10.1016/j.parco.2022.102918. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85126818076&doi=10.1016%2fj.parco.2022.102918&partnerID=40&md5=cd22c4a9982c5319ae878f965ed38485.

[103] Y. Ramdane et al. "SDWP: A New Data Placement Strategy for Distributed Big Data Warehouses in Hadoop". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11708 LNCS (2019), pp. 189–205. DOI: 10.1007/978-3-030-27520-4_14. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077122775&doi=10.1007%2f978-3-030-27520-4_14&partnerID=40&md5=6c06c7e95f92161f55679777e43ca142.

[104] Y. Ramdane et al. “SkipSJoin: A new physical design for distributed big data warehouses in hadoop”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11788 LNCS (2019), pp. 255–263. DOI: 10.1007/978-3-030-33223-5_21. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85076149464&doi=10.1007%2f978-3-030-33223-5_21&partnerID=40&md5=081e1cbda0c2dc6983b69f1ae6980b1f.

[105] F. Ravat and Y. Zhao. “Data Lakes: Trends and Perspectives”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11706 LNCS (2019), pp. 304–313. DOI: 10.1007/978-3-030-27615-7_23.

[106] G.M. Rocha, P.L. Capelo, and C.D.A. Ciferri. “Healthcare Decision-Making Over a Geographic, Socioeconomic, and Image Data Warehouse”. In: *Communications in Computer and Information Science* 1260 CCIS (2020), pp. 85–97. DOI: 10.1007/978-3-030-55814-7_7. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85090094843&doi=10.1007%2f978-3-030-55814-7_7&partnerID=40&md5=0d16817c8820c98c6784dfebe299c892.

[107] N. Rodriguez Brisaboa et al. “Semantrix: A compressed semantic matrix”. In: vol. 2020-March. 2020, pp. 113–122. DOI: 10.1109/DCC47342.2020.00019.

[108] S. Saebao, S. Matayong, and N. Trakulmaykee. “QoX based ETL Design for Business Intelligence System of Lecturers’ Qualifications Analysis”. In: 2020, pp. 539–542. DOI: 10.1109/ECTI-CON49241.2020.9158113. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091849886&doi=10.1109%2fECTI-CON49241.2020.9158113&partnerID=40&md5=30f53577aec1540c14dd7674bc318f7b.

[109] A. Sakka et al. “A volunteer design methodology of data warehouses”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11157 LNCS (2018), pp. 286–300. DOI: 10.1007/978-3-030-00847-5_21. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85054883737&doi=10.1007%2f978-3-030-00847-5_21&partnerID=40&md5=7a074641c54e8a7a17a981a583656949.

[110] N. Sanprasit, T. Titijaroonroj, and K. Kesorn. “A semantic approach to automated design and construction of star schemas”. In: *Engineering and Applied Science Research* 48.5 (2021), pp. 518–528. DOI: 10.14456/easr.2021.54. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111145740&doi=10.14456%2feasr.2021.54&partnerID=40&md5=22bc6d0571682e56c214288983b5b147.

[111] N. Sanprasit et al. “Intelligent approach to automated star-schema construction using a knowledge base”. In: *Expert Systems with Applications* 182 (2021). DOI: 10.1016/j.eswa.2021.115226. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107669008&doi=10.1016%2fj.eswa.2021.115226&partnerID=40&md5=925ad4c9c2884267f8b2aa090c27f342.

[112] L. Sautot, S. Bimonte, and L. Journaux. “A semi-automatic design methodology for (big) data warehouse transforming facts into dimensions”. In: *IEEE Transactions on Knowledge and Data Engineering* 33.1 (2021), pp. 28–42. DOI: 10.1109/TKDE.2019.2925621. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85097761649&doi=10.1109%2fTKDE.2019.2925621&partnerID=40&md5=2e7086a62d60e9089e5233806ec2ca2d.

[113] P.N. Sawadogo. “Textual Data Analysis from Data Lakes”. In: *Communications in Computer and Information Science* 1064 (2019), pp. 558–563. DOI: 10.1007/978-3-030-30278-8_54.

[114] P.N. Sawadogo, J. Darmont, and C. Noûs. “Joint Management and Analysis of Textual Documents and Tabular Data Within the AUDAL Data Lake”. In: *Lecture Notes in Computer Science (including subseries Lec-*

ture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12843 LNCS (2021), pp. 88–101. DOI: 10.1007/978-3-030-82472-3_8.

[115] P.N. Sawadogo et al. "Metadata Systems for Data Lakes: Models and Features". In: *Communications in Computer and Information Science* 1064 (2019), pp. 440–451. DOI: 10.1007/978-3-030-30278-8_43.

[116] J. Schering et al. "Potentials of Bicycle Infrastructure Data Lakes to Support Cycling Quality Assessment". In: vol. P-326. 2022, pp. 783–794. DOI: 10.18420/inf2022_66.

[117] Frank Schooneveldt. *Topicus Pension and Wealth*. Ed. by Topicus Finance. URL: https://topicus.nl/en/solutions/pension-and-wealth.

[118] Khurram Shahzad and Jelena Zdravkovic. "A goal–oriented approach for business process improvement using process warehouse data". In: *The Practice of Enterprise Modeling: Second IFIP WG 8.1 Working Conference, PoEM 2009, Stockholm, Sweden, November 18-19, 2009. Proceedings 2*. Springer. 2009, pp. 84–98.

[119] J. Singh, G. Singh, and B.S. Bhati. "The Implication of Data Lake in Enterprises: A Deeper Analytics". In: 2022, pp. 530–534. DOI: 10.1109/ICACCS54159.2022.9784986.

[120] Y. Song et al. "Design and construction of the data warehouse based on hadoop ecosystem at HLS-II". In: Joint Accelerator Conferences Website (JACoW), 2018, pp. 233–235. DOI: 10.18429/JACoW-PCaPAC2018-FRCB2.

[121] Helmut Spengler et al. "Improving data quality in medical research: A monitoring architecture for clinical and translational data warehouses". In: ed. by de Herrera A.G.S. et al. Vol. 2020-July. Institute of Electrical and Electronics Engineers Inc., 2020, pp. 415–420. DOI: 10.1109/CBMS49503.2020.00085.

[122] J. Sreemathy et al. "Data Validation in ETL Using TALEND". In: 2019, pp. 1183–1186. DOI: 10.1109/ICACCS.2019.8728420. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85067929993&doi=10.1109%2fICACCS.2019.8728420&partnerID=40&md5=83a3509441b6a848825cdc1e8673169e.

[123] J. Sreemathy et al. "Overview of ETL Tools and Talend-Data Integration". In: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1650–1654. DOI: 10.1109/ICACCS51430.2021.9441984.

[124] V.L. Takács et al. "Data warehouse hybrid modeling methodology". In: *Data Science Journal* 19.1 (2020), pp. 1–23. DOI: 10.5334/dsj-2020-038.

[125] S. Taktak et al. "Model-driven approach to handle evolutions of OLAP requirements and data source model". In: *Communications in Computer and Information Science* 880 (2018), pp. 401–425. DOI: 10.1007/978-3-319-94764-8_17. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85050372480&doi=10.1007%2f978-3-319-94764-8_17&partnerID=40&md5=ddd171f5601e87b0fae67ad7bb357a2a.

[126] M.A.C. Teixeira et al. "Data mart construction based on semantic annotation of scientific articles: A case study for the prioritization of drug targets". In: *Computer Methods and Programs in Biomedicine* 157 (2018), pp. 225–235. DOI: 10.1016/j.cmpb.2018.01.010. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041502663&doi=10.1016%2fj.cmpb.2018.01.010&partnerID=40&md5=572e949349b17560172926d8175872b2.

[127] Christian Thomsen et al. "Programmatic ETL". In: *Lecture Notes in Business Information Processing* 324 (2018). Ed. by Zimanyi E., pp. 21–50. DOI: 10.1007/978-3-319-96655-7_2.

[128] Topicus.Finance. *Fyndoo*. URL: https://fyndoo.com/.

[129] L. Toumi, A. Ugur, and Y. Azzi. "GPU-Based PSO for Bitmap Join Indexes Selection Problem in Data Warehouses". In: 2019. DOI: 10.1109/ICAEE47123.2019.9014703. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082174845&doi=10.1109%2fICAEE47123.2019.

9014703 & partnerID = 40 & md5 = e7fb23c43e3a0af322f4d15c023d431b.

[130] María Isabel Uvidia Fassler et al. "Moving towards a methodology employing knowledge discovery in databases to assist in decision making regarding academic placement and student admissions for universities". In: *International Conference on Technology Trends*. Springer. 2017, pp. 215–229.

[131] Y. Vlasiuk and V. Onyshchenko. "Data Mesh as Distributed Data Platform for Large Enterprise Companies". In: *Lecture Notes on Data Engineering and Communications Technologies* 181 (2023), pp. 183–192. DOI: 10.1007/978-3-031-36118-0_17.

[132] K. Wahyudi et al. "Business Intelligence for Employment Classification in Jakarta Government Data". In: 2019. DOI: 10.1109/ICISS48059.2019.8969851.

[133] A. Walha, F. Ghozzi, and F. Gargouri. "Design and Execution of ETL Process to Build Topic Dimension from User-Generated Content". In: *Lecture Notes in Business Information Processing* 415 LNBIP (2021), pp. 374–389. DOI: 10.1007/978-3-030-75018-3_25. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85111168430&doi=10.1007%2f978-3-030-75018-3_25&partnerID=40&md5=95c4a003ac4a86689c94855fab70f65a.

[134] J. Wang and B. Liu. "Design of ETL Tool for Structured Data Based on Data Warehouse". In: 2020. DOI: 10.1145/3424978.3425101. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85094915712&doi=10.1145%2f3424978.3425101 & partnerID = 40 & md5 = 9ee89dc6e78b67a98d62a9bc131ef5f4.

[135] Z. Wang et al. "Tempura: A general cost-based optimizer framework for incremental data processing". In: *Proceedings of the VLDB Endowment* 14.1 (2020), pp. 14–27. DOI: 10.14778/3421424.3421427. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85097296719&doi=10.14778%2f3421424.3421427 & partnerID = 40 & md5 = a6058ebfa0cef38aad614126168d1fdf.

[136] Z. Wang et al. "Tempura: a general cost-based optimizer framework for incremental data processing (Journal Version)". In: *VLDB Journal* 32.6 (2023), pp. 1315–1342. DOI: 10.1007/s00778-023-00785-1. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85150429848&doi=10.1007%2fs00778-023-00785-1&partnerID=40&md5=7300b85301cc43740dca00dd251b74b6.

[137] R. Wrembel. "Data Integration, Cleaning, and Deduplication: Research Versus Industrial Projects". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13635 LNCS (2022), pp. 3–17. DOI: 10.1007/978-3-031-21047-1_1. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85145008047&doi=10.1007%2f978-3-031-21047-1_1&partnerID=40&md5=285f18cc63bd66023f41297c18d3d668.

[138] Y. Yang et al. "Automatic Machine Learning-Based OLAP Measure Detection for Tabular Data". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13428 LNCS (2022), pp. 173–188. DOI: 10.1007/978-3-031-12670-3_15. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135888713&doi=10.1007%2f978-3-031-12670-3_15&partnerID=40&md5=78478d00e5079619b23dd9d133f6bb13.

[139] Yeisol Yoo and Jin Soung Yoo. "RFID data warehousing and OLAP with hive". In: Institute of Electrical and Electronics Engineers Inc., 2019, pp. 476–483. DOI: 10.1109/IUCC/DSCI/SmartCNS.2019.00105.

[140] Yue Yu et al. "Developing an ETL tool for converting the PCORnet CDM into the OMOP CDM to facilitate the COVID-19 data integration". In: *Journal of Biomedical Informatics* 127 (2022). DOI: 10.1016/j.jbi.2022.104002.

[141] A. Zekri et al. "Trajectory ETL modeling". In: *Smart Innovation, Systems and Tech-*

nologies 76 (2018), pp. 380–389. DOI: 10.1007/978-3-319-59480-4_38.

[142] M. Zekri, S.B. Yahia, and I. Hilali-Jaghdam. "A Software Prototype for Multidimensional Design of Data Warehouses Using Ontologies". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11684 LNAI (2019), pp. 273–284. DOI: 10.1007/978-3-030-28374-2_24. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072867181&doi=10.1007%2f978-3-030-28374-2_24&partnerID=40&md5=9589ded6b1248fe92c969767b3fe7081.

[143] M. Zekri, S. Zahaf, and F. Gargouri. "Characteristics of the decision-making dimension of the BPIS". In: vol. 164. 2019, pp. 285–291. DOI: 10.1016/j.procs.2019.12.185. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85079842636&doi=10.1016%2fj.procs.2019.12.185&partnerID=40&md5=8fdbb11e09f40a01364947c58de82055.

[144] M. Zekri, S. Zahaf, and F. Gargouri. "Specification of the data warehouse for the decision-making dimension of the Bid Process Information System". In: vol. 159. 2019, pp. 1190–1197. DOI: 10.1016/j.procs.2019.09.288. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85076259388&doi=10.1016%2fj.procs.2019.09.288&partnerID=40&md5=fd4b14583701755ade1370eb99dbc19e.

[145] Qianqian Zheng. "ETL Based Data Integration Scheduling". In: ed. by Subramanian K. Vol. 12509. SPIE, 2023. DOI: 10.1117/12.2655919.

A Overview of tools

Name	UI/Code	Git rating/community size	Paid option	Notable features
Airbyte	UI, Terraform,and API	13.2k stars, 863 contributors	Yes	DBT for transformations
Apache Airflow	UI and Python	33.5k stars, 2804 contributors	No	
Apache Beam	Python, Java, GO, typescript, Scala, SQL, YAML	7.4k stars, 1170 contributors	No	Multi-language pipelines
Apache Camel	Java	5.2k stars, 1029 contributors	No	Integration tool with ETL
Apache Druid	Web UI with SQL queries	13.1k stars, 591 contributors	No	
Apache Hadoop	MapReduce	14.2k stars, 573 contributors	No	Cluster computation
Apache Hive	CLI	5.3k stars, 372 contributors	No	
Apache Hop	GUI + web, CLI tools	813 stars, 70 contributors	No	Based on Pentaho
Apache Kafka	Java and Scala	26.9k stars, 1105 contributors	No	Real-time streaming
Apache NiFi	Web UI,	4.2k stars, 459 contributors	No	
Apache SeaTunnel	CLI	7k stars, 252 contributors	No	
Apache Spark	Python, Java, R, Scala, SQL	37.9k stars, 2042 contributors	No	Big data analysis
CloudQuery	CLI	5.4k stars, 142 contributors	Yes	Transformation with DBT
Dagster	Python and Web UI	9.7k stars, 380 contributors	Only paid	Integrates with DBT and Airbyte
DBT	CLI, SQL	8.5k stars, 290 contributors	Yes	Integrates well with other tools
Kestra	Localhost GUI with IDE	5.4k stars, 194 contributors	Yes	Integrates with Airbyte and DBT
Knime	GUI	144 stars, 17 contributors on GitHub	Yes	Has own community platform
Mage	GUI or own IDE, Python, R and SQL	6.6k stars, 92 contributors	No	
Meltano	CLI	1.5k stars, 120 contributors	No	Transformation with DBT
Pentaho Community edition	Low-code GUI	7.2k stars, 221 contributors	No	
Prefect	Python + monitoring UI	14.1k stars, 227 contributors	Yes	
PipelineWise	CLI	597 stars, 45 contributors	No	Requires Singer
Python libraries*	Code	NA	No	Scheduling with cron
R_etl	R	NA	No	Dedicated R package
Singer	CLI	∼1.5k stars, ∼30 contributors	No	PipelineWise scheduling and monitoring

Table 2: *The Python libraries include: Ethereum-etl, Luigi, Petl, Pygrametl

The programming language is mentioned if the tool uses code as its main way of building ETL pipelines. If the application is more low/no code, this is indicated in the UI used. Sometimes both options are possible while other times a UI is only for monitoring the pipelines, not for building.