The Journal of Systems and Software 226 (2025) 112391

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

NoSQL database education: A review of models, tools and teaching methods ☆

Nirnaya Tripathi

M3S Research Group, University of Oulu, Oulu 90014, Finland

ARTICLE INFO

Keywords:
NoSQL
Education
Teaching methods
Constructive alignment
Database
Systematic literature review
Software Engineering
Computer science

ABSTRACT

NoSQL databases are essential for managing modern data-intensive applications. While SQL education is a crucial part of the software engineering and computer science curriculum, it is insufficient in addressing the rise of big data and cloud infrastructures. Despite extensive research on SQL education, there is limited exploration of NoSQL education, particularly in teaching methods and data models. This study addresses this gap by conducting a systematic literature review on NoSQL database education, aiming to assess current research, teaching practices, models, tools, scalability, and security mechanisms while offering a framework for integrating NoSQL into academic curricula. Out of 386 articles, 28 were selected for detailed analysis, focusing on NoSQL teaching methods, models, and curriculum development. Findings revealed that document-oriented and graph databases, especially MongoDB, Cassandra, and Neo4j, are the most taught. The project-based learning approach was the most common teaching method. Challenges identified include adapting to technological advancements, addressing diverse student needs, and the shift to online learning. This review contributes valuable insights into NoSQL education and offers recommendations for improving teaching practices in software engineering curricula.

Contents

☆ Editor: Alexander Serebrenik.

E-mail address: nirnaya.tripathi@oulu.fi.

https://doi.org/10.1016/j.jss.2025.112391
Received 23 September 2024; Received in revised form 7 February 2025; Accepted 15 February 2025
Available online 25 February 2025

1. Introduction

Database systems education is an essential part of the software engineering curriculum as it equips future software engineers with the necessary knowledge and skills to design, develop, and manage data-intensive applications effectively. Traditional structured query language (SQL) database courses focus on teaching students concepts related to SQL relational databases, covering entities, relationships, and data structuring (Taipalus and Seppänen, 2020). However, with the rise of big data challenges, such as scalability, fixed schema, data modeling, and cloud-based infrastructures, there is a growing demand for students to learn about non-structured query languages (NoSQL), which are also commonly referred to as "Not Only SQL" databases (Bjeladinovic et al., 2020; Corbellini et al., 2017). NoSQL databases are nonrelational systems capable of handling large volumes of unstructured data and diverse data models, such as key–value, document, graph, and wide-column databases, in contrast to the consistent relational model of SQL databases (Nayak et al., 2013; Han et al., 2011) (also see Fig. 1). Just like SQL database education, NoSQL database education is crucial because students will likely encounter NoSQL databases in their professional careers. Major enterprises are now transitioning to NoSQL databases for fast data access and to address the limitations of relational databases (Chen and Lee, 2019). Although relational databases remain powerful tools, current trends suggest that they will not be the only databases in use (Wang and Wang, 2023; Zinovieva et al., 2021). Given the surge in big data and cloud processing capabilities, integrating NoSQL databases into academic curricula is crucial to equip students for future database scenarios. This is important for the following reasons:

- Teaching students about NoSQL databases provides them with the skills to efficiently manage and analyze vast and diverse data, offering flexible data modeling options (Vera-Olivera et al., 2021).
- NoSQL databases are designed for horizontal scalability, making them well suited for modern software application development. They enable students to learn how to work with rapidly growing data volumes, including real-time analytics, recommendation engines, and Internet of Things systems (Bjeladinovic et al., 2020).

1.1. Research problem and objectives

Many studies have been published regarding SQL education, as mapped in Taipalus and Seppänen (2020). Additionally, NoSQL literature has also been reviewed in studies by Martinez-Mosquera et al. (2020) and Ahmed et al. (2018) (refer to Section 2.3). Available evidence suggests that there are existing reports on NoSQL database education. However, there is a dearth of comprehensive literature survey research focusing on NoSQL database education and its implications for teaching, data modeling, and management. In addition, the author of the current study has been teaching NoSQL database courses for several years and thus has a personal motivation to explore the topic. This lack of a thorough survey of NoSQL academic studies and the author's personal motivation underscore the importance of understanding NoSQL technologies and their teaching methodologies in higher education, highlighting the need for a literature review study designed to accomplish the following objectives:

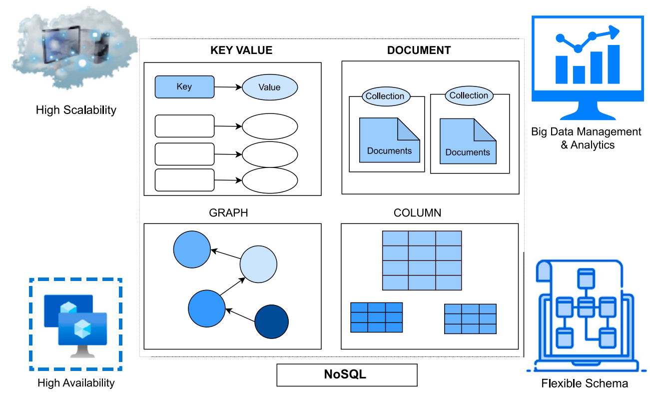

Fig. 1. NoSQL models and benefits.

- To assist instructors and course developers in assessing existing knowledge in the field of NoSQL database management
- To examine the current understanding of NoSQL models and tools, as well as the utilization of NoSQL topics, such as data modeling techniques, scaling, and security mechanisms
- To offer evidence-based teaching practices and a foundational teaching framework for future educators and students in the field of database education

1.2. Research process

To accomplish the research objectives, the author conducted a systematic literature review (SLR) (refer to Fig. 3) by formulating four research questions (RQs) (see Table 1). The author searched for research articles related to NoSQL education in peer-reviewed scientific databases, such as IEEE, Scopus, ACM, and Web of Science. The search process also involved snowballing to gather additional literature. A total of 386 sources were identified. After the selection process, 32 articles were recognized as primary study candidates (refer to 4), six of which were found through the snowballing technique. With these 32 primary study articles, the current study aims to systematically organize and analyze the literature on NoSQL education in order to address the four RQs.

1.3. Research contributions

This study aims to make the following contributions through its research findings:

- Provide a comprehensive overview of scholarly information pertaining to this field over time (Section 4)

- Offer insights into various NoSQL database models and tools (e.g., MongoDB, Cassandra, and Neo4j), as well as common topics covered in higher education settings (Section 5)
- Provide valuable insights into effective pedagogical teaching methods and best practices for NoSQL education (see Section 6)
- Identify key challenges in NoSQL education (Section 6.5), such as addressing the diverse backgrounds of students and adapting to new learning environments (i.e., online and blended learning)

1.4. Paper overview

The paper is structured into multiple sections. Section 2 outlines the background and related work. Section 3 details the research methodology employed for the literature review, while Sections 4, 5, and 6 present our findings on NoSQL education. In Section 7, we discuss topics related to NoSQL versus teaching methods, provide recommendations for educators, and assess the validity of the study. Finally, the conclusions are provided in the final section.

2. Background literature

2.1. Big data and NoSQL

The term "big data" refers to extensive and complex datasets that present challenges for traditional processing applications. It encompasses both structured and unstructured data from various sources (Corbellini et al., 2017). In response to the rapid growth of data types and the need for scalability, high availability, and fault tolerance, which traditional relational databases struggle to meet, NoSQL databases have emerged. These databases, characterized by a schemaless design and eventual consistency, provide solutions through horizontal scalability across numerous servers. These databases are grouped into types such as key–value, column, document oriented, and graph, each with unique features suited for specific applications (Nayak et al., 2013). Nonstructured query language databases support different data models and lack a standardized modeling approach. Their query languages and scalability make them suitable for handling large volumes of structured, semistructured, and unstructured data, challenging the previously dominant role of SQL-based relational databases in data management and database security (Han et al., 2011). This study explores various topics related to NoSQL, such as core concepts, types of NoSQL databases, data modeling, query languages, scaling and performance optimization, data management, and database security, in the context of education, as illustrated in the study framework in Fig. 2.

2.2. NoSQL education

The significance of NoSQL education is underscored by the limitations of SQL databases in handling large, unstructured datasets. This has led to the emergence of NoSQL technologies as crucial alternatives for managing big data and meeting modern application requirements. The lack of NoSQL teaching cases for undergraduate courses, as mentioned in Bjeladinovic et al. (2020), Wang and Wang (2023), and Zinovieva et al. (2021), highlights a substantial gap in resources that elaborate on NoSQL's role in handling unstructured big data, thereby emphasizing the necessity for comprehensive educational materials. Within this context, the work by Stanier (2012), Bjeladinovic et al. (2020), Wang and Wang (2023), and Zinovieva et al. (2021) provides valuable insights into integrating NoSQL concepts into academic curricula. This includes practical examples using MongoDB, innovative teaching methods, and the integration of massive open online courses (MOOCs) for teaching NoSQL topics.

In the context of higher education, constructive alignment is a teaching framework designed to ensure effective and meaningful learning outcomes for students in university education (Biggs et al., 2022).

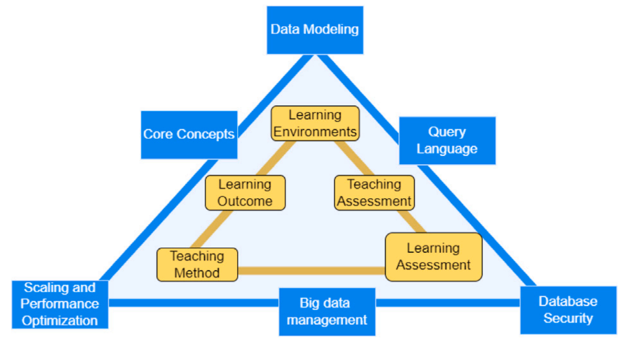

Fig. 2. The study framework includes a breakdown of NoSQL topics and teaching principles for NoSQL education.

Table 1
Research Questions.

Research questions	Rationale
RQ1. What are the trends and publication patterns in NoSQL education research?	This question aims to analyze the distribution of NoSQL education literature over different years, sources, paper contribution types, and student types and sizes.
RQ2. What NoSQL models, topics, and tools are most taught in higher education settings?	This question examines the types of NoSQL models (document, key–value, column, and graph databases) and tools (e.g., MongoDB, Neo4j, and Cassandra) covered in courses, as well as key NoSQL database topics.
RQ3. What types of pedagogical approaches are used in teaching NoSQL databases in higher education?	This question uncovers innovative approaches and best practices by examining the teaching methods employed in NoSQL education.
RQ4: What are the key challenges in teaching NoSQL databases?	This question aims to identify the challenges outlined in the discussion on NoSQL education.

Evident in SE education Cain and Babar (2016), Hynninen et al. (2018), Askarbekuly et al. (2021), this approach involves aligning different course components, such as learning objectives, teaching methods, assessment tasks, and feedback mechanisms, to create a cohesive and purpose-driven learning experience for students. To explore pedagogical approaches in NoSQL education, we will consider aspects of the learning environment, learning outcomes, teaching and learning methods, and learning and teaching assessments, as depicted in the study framework (see Fig. 2).

2.3. Related literature review studies

The following SLR research highlights the growing importance of big data modeling, management, and the role of NoSQL databases. Martinez-Mosquera et al. (2020) conducted an SLR on big data modeling and management, finding a strong emphasis on entity relationship and document-oriented models at the conceptual and logical levels, with MongoDB being the most used implementation. Ahmed et al. (2018) focused on NoSQL databases for big data processing, addressing structural issues and real-time data mining techniques, and proposed future research directions and challenges. Similarly, Khan et al. (2023) discussed the significance of software architecture in big data processing, particularly for SQL and NoSQL databases, aiming to address cloud data portability, interoperability, and the comparative capabilities of SQL and NoSQL databases. Vera-Olivera et al. (2021) concentrated on modeling NoSQL databases, finding a preference for conceptual and logical levels over physical representation, with adaptations of UML and ER models for NoSQL databases. Taipalus and Seppänen (2020) aimed to consolidate teaching practices and research findings

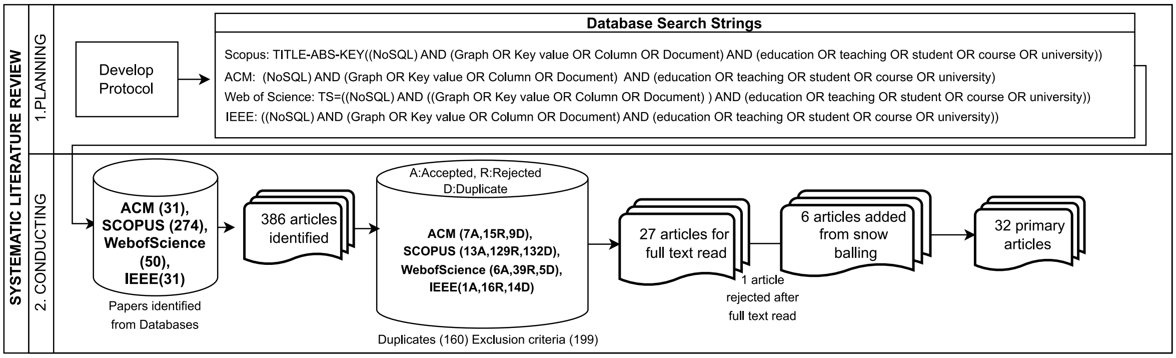

Fig. 3. The research process.

Table 2
Inclusion and Exclusion Criteria.

Criteria	Reference
I1. The article primarily addresses NoSQL education.	
I2. The article was published in conferences or journals.	
E1. The article is not in English, or its full text is unavailable.	
E2. The article is a literature review study.	Taipalus and Seppänen (2020)
E3. The article does not discuss the teaching of NoSQL in detail.	Menzin et al. (2020), Xiong et al. (2015), Pasterk and Bollin (2017)

through a systematic mapping study, uncovering six main themes in SQL education. Together, these related studies underscore the need for SLRs in NoSQL education to consolidate existing research, identify educational gaps, and outline effective teaching strategies.

3. Methodology

The research follows the guidelines established by Kitchenham et al. (2007) and employs the SLR tool Parsifal[1] as it helps in planning, conducting, and reporting the review. This study is methodically divided into three principal phases: planning, execution, and reporting. These phases collectively examine the pedagogical strategies, educational content, and breadth of NoSQL database topics integrated within diverse educational domains concerning database studies.

3.1. Planning

As highlighted in Section 2.3, there is a notable gap pertaining to the review of educational studies on NoSQL databases in the literature. This review seeks to explore the present understanding of NoSQL instruction, encompassing data models, data modeling methods, scaling and security strategies, and the pedagogical methodologies employed overall.

Research questions. Four RQs were formulated to address the study's objectives (refer to Table 1). The review protocol for this study was developed with Parsifal during the planning phase. In the finalization of the review protocol, thorough attention was given to various aspects, such as the keywords and inclusion criteria. The inclusion selection criteria are provided in Table 2. The search string structure in terms of population and intervention in keywords and synonyms is detailed below.

- Population: NoSQL AND (Graph OR Key value OR Column OR Document)
- Intervention: Education, course, teaching, university

A pilot study was undertaken to investigate these RQs. Initially, a set of keywords was used by the author to search two electronic databases, Scopus and Web of Science (WOS). However, the search yielded limited results in terms of peer-reviewed literature, indicating a lack of exploration of NoSQL education in academic studies. Consequently, the author decided to broaden the research by exploring sources such as research articles related to NoSQL education in databases (i.e., IEEE and ACM). The choice of these electronic databases was based on their wide usage in the software engineering domain and their support for advanced query searches.

3.2. Conducting the searches

The author conducted searches across four major digital databases without restricting the search by publication date or type. The specific search terms used and the results obtained from each database are detailed in the database search string in Fig. 3. In total, search efforts led to the identification of 386 relevant studies, with 31 from the ACM Digital Library, 31 from IEEE Xplore, 50 from Web of Science, and 274 from Scopus. The process used for selecting studies is depicted in the same figure.

Study selection and assessment. All 386 articles' meta-information was imported into Parsifal to identify primary studies. After 160 duplicate studies were eliminated, 226 articles remained for further assessment. The titles, abstracts, and keywords of the selected studies were reviewed, with additional reading of the introduction and conclusion when necessary. The inclusion and exclusion criteria were then applied to the title and abstract of each paper, reducing the pool of papers for full-text reading to 27. The excluded papers are listed in Table 2. Whenever there was uncertainty about a paper, it was reviewed again until a decision was made regarding its inclusion or exclusion and how it fit into the data extraction categories. The comparison and evaluation of these studies were carried out twice, resulting in the selection of 26 primary studies. One paper was rejected during this process (Taipalus, 2019). Snowballing was carried out by reading the references of primary papers to further investigate the topic, and this process led to the discovery of six relevant papers from Google Scholar (GScholar). Subsequently, 32 papers were selected as primary studies. An overview of the articles retrieved and selected from the four database sources can be seen in Fig. 4. In this study, quality assessment criteria were

[1] Parsifal https://parsif.al/.

Table 3
Extracted Data Classification.

Data Extraction classification	Section and RQs
Publication year, data source	Section 4.1, RQ1
Article contributions, education level, and students' size	Sections 4.2, 4.3
NoSQL models, tools, and topics	Section 5, RQ2
Teaching approach (Environment, methods, and assessments)	Section 6, RQ3
NoSQL education challenges	Section 6.5, RQ4

used specifically in relation to the pertinence of the articles to NoSQL education. The pertinence aspect of the articles was examined by classifying them as full, partial, or marginal. Full fully focuses on NoSQL education, partial presents knowledge and discusses NoSQL education but does not fully focus on it, and marginal has a minor or very limited focus on NoSQL education. These assessment criteria were applied to an included set of 32 primary studies. The list of primary studies with their pertinent assessment scores is found in Table 4.

3.3. Analysis

The data extraction and analysis were performed using the deductive thematic approach (Cruzes and Dyba, 2011). A data extraction form, as shown in Table 3, was employed to extract data from the primary studies. Prior to the actual data extraction, a pilot extraction process was conducted. The primary articles were then imported into the NVivo data analysis tool to code the data for deductive analysis. Qualitative data from the primary articles were analyzed to address our RQs and achieve the study's objectives. The author utilized a deductive coding approach to develop codes for extracting pertinent information from the data. These codes encompassed NoSQL focus (single, mixed, and hybrid), model types (column, graph, document, and key–value databases) and tools, and NoSQL topics. Moreover, data on teaching approaches regarding the teaching environment (face-to-face [F2F], hybrid, and online learning environments), teaching methods, assessment techniques, and NoSQL education challenges were extracted. The author reviewed the extracted data from the primary studies multiple times to identify all pertinent texts relevant to the RQs. Some figures were created using Excel, while others were made using hand-drawn methods.

4. Bibliometric analysis and trends

To answer the first RQ, we analyze the distribution of NoSQL education literature across different years, sources, paper contribution types, and student types and sizes. These details are discussed below.

4.1. Publication trend

In Fig. 4, we can observe the distribution of relevant NoSQL education articles across different databases and time frames, along with the articles per source, the accepted articles, and the publications per year. As indicated in the figure, 71% (274) of the papers were from Scopus, while 13% were from Web of Science. The remaining databases each contained 8% of the papers. Among the accepted articles, 12 papers were from Scopus, while 8 and 7 papers were from Web of Science and the ACM Digital Library, respectively. In addition, six papers were derived from Google Scholar through the snowball technique.

The distribution of the publication year in Fig. 5 shows a peak in research activity in 2016, with 7 studies published, making it the most productive year. Recent years (2023–2024) also show significant contributions, with a combined total of 9 studies, reflecting ongoing interest in the topic. Earlier years (2013–2015) show minimal activity, indicating the gradual emergence of research focus in this area. Overall, the trend suggests steady growth in research output, with notable spikes in certain years. Furthermore, most studies (23) have a Full

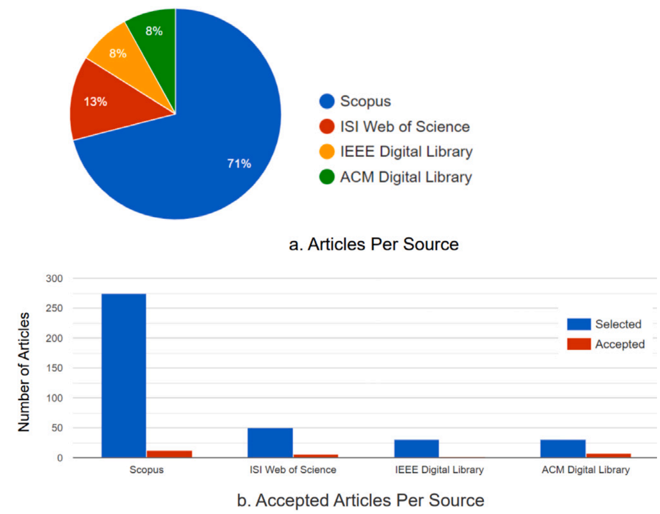

Fig. 4. Overview of the search and selection of articles per database source.

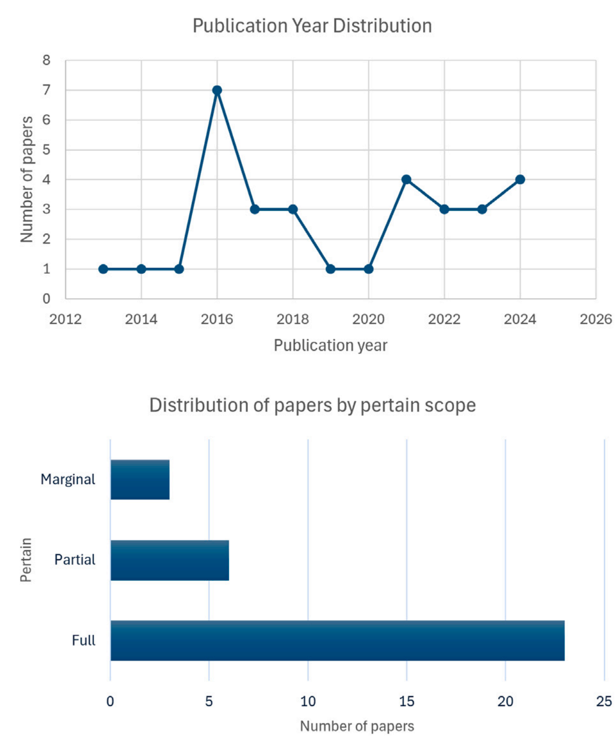

Fig. 5. The number of articles published each year and their study pertain related to NoSQL education.

pertain related to NoSQL education as seen in Fig. 5, reflecting comprehensive exploration, while fewer have Partial pertain (6) or Marginal pertain (3). This shows a strong focus on in-depth research across the NoSQL education studies. Furthermore, frequent key concepts related to NoSQL education, mentioned from primary sources, is presented in Fig. 6.

4.2. Contribution type

The primary research articles were classified as experience reports and empirical evaluations. Experience report articles offer practical insights and strategies for educators, while empirical evaluation studies provide evidence-based conclusions from students. Table 5 displays the contribution types of papers.

Table 4
Overview of Primary Studies.

ID	Ref	Year	Pertain	Source	Model type	ID	Ref	Year	Pertain	Source	Model type
P1	Wang and Wang (2023)	2023	Partial	Scopus	Key–value, Column, Document, Graph	P17	Poulová et al. (2016)	2016	Marginal	WOS	NoSQL
P2	Fowler et al. (2016)	2016	Full	Scopus	Document	P18	Byrne et al. (2017)	2017	Full	WOS	Document, SQL
P3	Mingyu et al. (2017)	2017	Partial	Scopus	Key Value, Document, SQL	P19	Li et al. (2016)	2016	Full	WOS	Document, Graph, SQL
P4	Mitri (2023)	2023	Full	Scopus	Key Value	P20	Dong and Wang (2016)	2016	Full	WOS	Key Value, Column, Document, Graph
P5	Werner (2022)	2022	Full	Scopus	Key value, Column, Document, Graph, SQL	P21	Guo et al. (2016)	2016	Full	IEEE	Document, SQL
P6	Grillenberger and Romeike (2014)	2014	Full	Scopus	NoSQL	P22	Alkhabaz et al. (2021)	2021	Full	ACM	Document, Graph, SQL
P7	Wu (2021)	2021	Partial	Scopus	Key value, Document, SQL	P23	German et al. (2016)	2016	Marginal	ACM	Document
P8	Krasheninnik et al. (2022)	2022	Partial	Scopus	Document	P24	Wu (2023)	2023	Partial	ACM	Document, SQL
P9	Chen et al. (2021)	2021	Full	Scopus	Document, Graph, SQL	P25	Sattar et al. (2013)	2013	Full	ACM	Document, SQL
P10	Werner and Bach (2018)	2018	Full	Scopus	Document	P26	Howard (2015)	2015	Marginal	ACM	Document
P11	Alawini et al. (2022)	2022	Full	Scopus	Document, Graph, SQL	P27	Kim (2020)	2020	Full	ACM	Column, Document
P12	Zinovieva et al. (2021)	2021	Full	Scopus	Document	P28	Silva et al. (2016)	2016	Partial	ACM	Key value, Document, SQL
P13	Hwang (2018)	2018	Full	WOS	Graph	P29	Kotsifakos et al. (2019)	2019	Full	GScholar	Graph
P14	Mohan (2018)	2018	Full	WOS	Key Value, Document, Graph	P30	Hiles and Agha (2017)	2017	Full	GScholar	Document
P15	Ehlers (2024)	2024	Full	GScholar	Document, Graph, SQL	P31	Meyer et al. (2024)	2024	Full	GScholar	Column, Document, Graph, SQL
P16	Amparo and Huang (2024)	2024	Full	GScholar	Document, SQL	P32	Shreevatsan et al. (2024)	2024	Full	GScholar	Document

Table 5
Contribution Type.

Aspect	Reference	Total
Experience report	Grillenberger and Romeike (2014), Poulová et al. (2016), Byrne et al. (2017), Li et al. (2016), Hwang (2018), Guo et al. (2016), Sattar et al. (2013), Howard (2015), German et al. (2016), Hiles and Agha (2017), Mingyu et al. (2017), Shreevatsan et al. (2024), Alawini et al. (2022), Silva et al. (2016), Amparo and Huang (2024)	15 (47%)
Empirical Evaluation	Alkhabaz et al. (2021), Chen et al. (2021), Dong and Wang (2016), Fowler et al. (2016), Kim (2020), Kotsifakos et al. (2019), Krasheninnik et al. (2022), Mitri (2023), Mohan (2018), Wang and Wang (2023), Werner and Bach (2018), Werner (2022), Wu (2021), Wu (2023), Zinovieva et al. (2021), Ehlers (2024), Meyer et al. (2024)	17 (53%)

Fig. 6. Key concepts frequently associated with NoSQL education, as seen in primary sources.

Experience reports. The articles in this category were organized according to their focus on qualitative insights provided by educators and cover 15 primary sources (47%). These articles feature experiences in innovative pedagogical strategies, reflections on teaching practices, and challenges encountered in the education sector. For example, Alawini et al. (2022) reflected on their innovative pedagogical system that aids students in learning the schema and query languages of NoSQL databases. Similarly, the authors in Byrne et al. (2017) expressed their views on the importance of data model normalization in database education by comparing the data structures of a traditional relational database with those of a document-based database at the logical and physical database design stage. Another experience study was Grillenberger and Romeike (2014), which described the challenges that computer science education needs to deal with when incorporating NoSQL developments into teaching. Finally, narrative insights regarding the teaching security issues of NoSQL databases were provided in Guo et al. (2016).

Empirical evaluation. This category comprised papers that systematically investigate educational phenomena, often using quantifiable data and providing data-driven evidence collected from students and cover 17 primary sources (53%). For example, in Alkhabaz et al. (2021), an in-depth analysis of students' MongoDB homework assignments

Table 6
Education Type and Size.

Aspect	Student size <= 100	Student size > 100	N/A
Under Graduate	Mohan (2018), Kim (2020)	Krasheninnik et al. (2022), Ehlers (2024)	Poulová et al. (2016)
Graduate	–	Wu (2021), Wu (2023)	Li et al. (2016)
Both	–	Alkhabaz et al. (2021), Chen et al. (2021)	Fowler et al. (2016)
N/A	Poulová et al. (2016), Li et al. (2016), Wang and Wang (2023)	Werner (2022)	–

was conducted to explore the errors made by students in a database course. Likewise, a quantitative analysis of student homework submissions was carried out in Chen et al. (2021) to explore students' learning processes while solving graph database problems. Students' survey feedback was gathered in Dong and Wang (2016), Wang and Wang (2023), and Mohan (2018) to evaluate NoSQL databases labs, knowledge of the basics of NoSQL among students, and the course NoSQL learning model.

4.3. Education level and student size

As indicated in Table 6, the primary articles discussed the offering of courses at the graduate and/or undergraduate level to serve distinct academic and professional objectives. Undergraduate courses aim to provide a foundational understanding of a discipline, while graduate courses emphasize specialized areas, fostering expertise and research. For instance, the Database Systems course with a NoSQL teaching case, as described in Alkhabaz et al. (2021) and Fowler et al. (2016), was predominantly taken by graduate students but was also open to undergraduates who had completed prerequisite programming and data structures courses. Conversely, Wu (2021, 2023) focused on offering a comprehensive overview of modern data management to graduate students.

Regarding class sizes, studies such as Wang and Wang (2023) and Mohan (2018) concentrated on courses with less than 100 students and those that created an ideal environment for active learning, personalized instruction, and close interaction between students and instructors. Conversely, the courses discussed in Chen et al. (2021) and Werner (2022) had more than 100 students and are generally regarded as large classes, which are common in many undergraduate programs, particularly in introductory courses. Managing and teaching in such settings present unique challenges and opportunities.

Answer to RQ1. Based on the above observation, the trends and publication patterns in NoSQL education research reveal consistent growth over the past decade, with a significant portion of studies sourced from Scopus, accounting for 71% of the papers. The research focuses on an equal distribution of contributions: experience reports, which offer practical teaching insights, and empirical evaluations, which provide data-driven assessments of student learning. These studies target both undergraduate and graduate students, with course sizes ranging from small groups to large classes, each presenting unique teaching opportunities and challenges.

5. NoSQL education

This section explores the findings for the second question, which examines the types of NoSQL models and databases covered in courses, as well as the NoSQL database topics addressed.

5.1. NoSQL focus

As shown in Tables 4 and 7, the primary articles concentrated on instructing a single NoSQL model, a blend of NoSQL models, or the integration of both NoSQL and SQL. The objective was to thoroughly encompass the distinctive features, advantages, and real-world uses of these data models.

Single NoSQL model. Single NoSQL data model is a critical component of NoSQL courses, as it provides students with a focused understanding of how a specific model, such as document, key–value, columnar, and graph databases, address specific challenges in handling unstructured and semistructured data. Several studies have leveraged single-model approaches to convey foundational NoSQL concepts effectively. Fowler et al. (2016) demonstrated the value of real-world projects, where students constructed a document database using CouchDB for a social media application. This hands-on experience allowed students to understand the practical role of document databases in contemporary applications. Similarly, Hiles and Agha (2017) focused on introducing MongoDB concepts to software professionals through straightforward, accessible teaching methods, emphasizing the simplicity and adaptability of document databases for varied use cases. Graph databases also featured prominently in NoSQL education, as illustrated by Kotsifakos et al. (2019). Their course used Neo4j and Cypher functions to introduce students to graph database commands, highlighting the relevance of graph models for complex relationships and connected data, such as social networks or recommendation systems.

Focusing on single NoSQL models offers students a deep understanding of specific architectures, such as document, key–value, or graph databases, and simplifies the learning process. Hands-on projects, like designing a CouchDB social media application (Fowler et al., 2016) or using Neo4j for graph queries (Kotsifakos et al., 2019), provide real-world relevance and practical skills. This approach may limit exposure to the broader NoSQL landscape, trade-offs between models, and polyglot persistence, where multiple models are combined. Balanced knowledge of database models prepares students for industry needs, especially polyglot persistence (Kim, 2020). A mixed or hybrid model course (see next section) can bridge these gaps.

Mixed NoSQL data models. Mixed NoSQL models in courses highlight the versatility and performance optimization achievable by combining the strengths of different data models, such as document stores, key–value pairs, column stores, and graph databases. This approach is critical for preparing students to tackle complex, real-world applications where diverse data requirements necessitate polyglot persistence (Dong and Wang, 2016), (Kim, 2020). For example, the course in Mohan (2018) introduced students to key–value, document, and graph models, providing a comprehensive understanding of their integration. Similarly, Dong and Wang (2016) incorporated all four NoSQL models in a cloud computing environment, enabling students to handle large, high-velocity datasets with varying structures. Such mixed-model approaches allow students to grasp the benefits of combining models, such as improved scalability, optimized query performance, and enhanced data representation. By teaching mixed NoSQL models, educators can provide students with a diverse skill set to design and implement versatile, high-performance NoSQL data systems (Kim, 2020; Werner, 2022).

Hybrid data models (NoSQL and SQL). Hybrid NoSQL models, which integrate both NoSQL and SQL databases, is vital for providing students with a comprehensive understanding of relational and nonrelational systems and their unique strengths and use cases (Chen et al., 2021; Li et al., 2016; Sattar et al., 2013; Silva et al., 2016). Courses such as those in Chen et al. (2021) and Li et al. (2016) incorporated document and graph databases alongside SQL, enabling students to grasp the comparative benefits of different paradigms. For instance, the work in Alawini et al. (2022) introduced schema and query languages across MySQL, Neo4j, and MongoDB, helping students understand how relational and nonrelational databases address diverse data requirements.

Table 7
NoSQL Models and Databases.

Source	Key value	Column	Document	Graph	SQL	Type
Hiles and Agha (2017)			MongoDB			Single
Werner and Bach (2018)			MongoDB, CouchDB			Single
Byrne et al. (2017)			MongoDB			Single
Fowler et al. (2016)			CouchDB			Single
Kotsifakos et al. (2019)				Neo4J		Single
Krasheninnik et al. (2022)			MongoDB			Single
Mitri (2023)	DynamoDB					Single
Hwang (2018)				Sap Hana		Single
Dong and Wang (2016)	Redis	Cassandra	MongoDB	Neo4j		Mixed
Kim (2020)		Cassandra	MongoDB			Mixed
Mohan (2018)	Redis		MongoDB	ArangoDB		Mixed
Werner (2022)	Hypertable	Cassandra	MongoDB, CouchDB	ArangoDB, Neo4j		Mixed
Alawini et al. (2022)			MongoDB	Neo4J	MySQL	Hybrid
Alkhabaz et al. (2021)			MongoDB	Neo4J	MySQL	Hybrid
Chen et al. (2021)			MongoDB	Neo4J	MySQL	Hybrid
Li et al. (2016)			MongoDB		SQLite	Hybrid
Mingyu et al. (2017)	BigTable		MongoDB		Oracle	Hybrid
Sattar et al. (2013)			MongoDB		Oracle	Hybrid
Silva et al. (2016)	Hbase		MongoDB		SQL	Hybrid
Wu (2021)	Hadoop, Spark		MongoDB		MySQL	Hybrid
Wu (2023)			MongoDB		MySQL	Hybrid
Guo et al. (2016)			MongoDB		SQLite	Hybrid
Amparo and Huang (2024)			MongoDB		MySQL	Hybrid
Ehlers (2024)			MongoDB	Neo4j	MariaDB	Hybrid
Meyer et al. (2024)		Cassandra	MongoDB	Neo4j	PostgreSQL	Hybrid

Similarly, Alkhabaz et al. (2021) and Chen et al. (2021) covered essential topics like data models, query languages, database design, and implementation in both SQL and NoSQL contexts. The advantages of teaching hybrid models enable students to create systems that combine the strengths of both database types for enhanced performance.

5.2. NoSQL model and database type

When instructing about NoSQL models, such as document stores, key–value pairs, column stores, and graph databases, as well as the database technologies associated with them, delineating their unique characteristics and specific applications is crucial. In the following section, we discuss the approach for each data model type. Detailed information is found in Table 7.

Column. Teaching column-family databases in NoSQL courses is crucial for helping students understand their unique architecture, performance advantages, and specific use cases. Unlike relational databases and other NoSQL types, column-family databases organize data by columns rather than rows, enabling optimized reading and writing of large datasets (Byrne et al., 2017). Courses such as those in Sattar et al. (2013) introduced column-oriented databases as a key NoSQL model, highlighting their role in handling high-throughput workloads and unstructured data like images and text. In the article (Wang and Wang, 2023), column-family databases were presented as more flexible than traditional relational systems, particularly for managing complex, large-scale datasets. Hands-on activities, such as those in Kim (2020), provided students with experience using Apache Cassandra, including its architecture, data modeling, and query language, demonstrating how column-oriented schemas can support efficient data distribution, replication, and querying. By integrating column-family databases into the curriculum, educators equip students with the skills to design and manage systems that excel in high-performance, data-intensive environments.

Key–value. Understanding key–value databases in NoSQL courses is essential for grasping their design principles, which emphasize schema flexibility, scalability, and performance in managing large data volumes (Alawini et al., 2022; Fowler et al., 2016). Courses such as those in Mingyu et al. (2017) and Fowler et al. (2016) emphasized hands-on experience with key–value databases like Redis and Amazon DynamoDB, enabling students to grasp core operations such as data storage, retrieval, and manipulation. These practical exercises help students see how key–value databases excel in simplicity and speed, particularly for use cases like caching, session management, and real-time analytics (Alawini et al., 2022), Kim (2020). A key aspect of these courses is comparing key–value models with other NoSQL types (document, column-family, and graph) and relational databases, as highlighted in Grillenberger and Romeike (2014) and Howard (2015). This comparative analysis allows students to understand the specific strengths and limitations of key–value databases, such as their exceptional performance for straightforward lookups and writes but limited support for complex querying and data relationships.

Document. Document databases in NoSQL courses is fundamental due to their flexibility, scalability, and ability to handle hierarchical and semistructured data. Document-oriented models, such as those using JSON or XML, enable schema-less data structures that adapt dynamically, as highlighted in Alawini et al. (2022), Alkhabaz et al. (2021), and Werner (2022). These courses, often centered around MongoDB, emphasize the advantages of schema-less designs for applications with evolving data requirements. Articles like Byrne et al. (2017) illustrate the suitability of document databases for semistructured data scenarios, such as content management and IoT systems. Similarly, German et al. (2016) and Wang and Wang (2023) demonstrate how document models enable scalable web application development by providing dynamic data modeling capabilities. Comparative analyses in Werner and Bach (2018) and Silva et al. (2016) between document and relational databases help students practice NoSQL and SQL queries, showcasing the performance benefits and flexibility of document-oriented approaches. Teaching document models equips students with the skills to design and implement scalable, flexible data solutions while understanding their limitations in specific use cases.

Graph. Teaching graph databases in NoSQL courses is crucial for equipping students to manage and query complex, interconnected data efficiently, especially in domains like social networks, logistics, spatial analysis, and natural language processing (Kotsifakos et al., 2019). Practical exercises detailed in Kotsifakos et al. (2019) and Werner (2022) provided hands-on experience with Neo4j, enabling students to create and manipulate nodes and relationships using Cypher query language. Comparative studies in Wu (2021) highlighted the strengths of graph databases over other data models, such as their efficiency in handling complex queries involving relationships. Courses like those described in Hwang (2018) introduced hybrid systems, such as SAP HANA, which integrate graph processing with relational databases, exposing students to cutting-edge technologies.

5.3. NoSQL topics in education

In education, the study of NoSQL encompasses essential topics, such as the core concepts and types of NoSQL databases, data modeling, query languages and create, read, update, and delete (CRUD) operations, scaling and performance optimization, big data, and security. These subjects lay the groundwork for a robust curriculum on NoSQL databases, equipping students with the skills to address real-world complexities in managing and processing large-scale data.

Core concepts and data models. Teaching core concepts and data models in NoSQL courses is vital for understanding NoSQL's emergence, its applications, and the limitations of traditional RDBMSs that it addresses (Dong and Wang, 2016). Mohan (2018) laid the groundwork by introducing NoSQL data models—document, key–value, wide-column, and graph—through illustrative examples, enabling students to understand the operational mechanisms and scenarios where each model excels. Wang and Wang (2023) highlighted the rise of NoSQL technologies, particularly emphasizing their importance for managing unstructured and semistructured data in the digital age. Wu (2021) provided a comparative perspective, examining NoSQL databases and their integration with big data technologies like Hadoop MapReduce and Spark, offering students insights into their respective strengths. By focusing on core concepts and data models, educators ensure students gain both theoretical knowledge and practical skills for addressing real-world data challenges.

Data modeling. Teaching data modeling in NoSQL courses is crucial for helping students design effective and efficient database solutions tailored to diverse application requirements. NoSQL systems require data modeling approaches that differ significantly from traditional relational methods, and several studies have highlighted their importance (Byrne et al., 2017). For example, Chen et al. (2021) focused on modeling and querying relationships using the Neo4j graph database and Cypher query language, emphasizing the nuances of graph-based pattern matching. Similarly, Alawini et al. (2022) explored schema modeling in Neo4j and MongoDB, offering students insights into comparative modeling strategies and the flexibility of schema-less and graph-based systems. Byrne et al. (2017) contrasted relational database design stages with MongoDB's schema design, while Hiles and Agha (2017) used MongoDB for practicing conceptual database design with constraint-based modeling. Kim (2020) emphasized Cassandra's query-driven data modeling approach, showing students how access patterns shape schema design in column-family databases. Mohan (2018) provided comprehensive exercises combining traditional techniques, like entity-relationship diagrams and normalization, with NoSQL-specific considerations, such as data aggregation and denormalization. While teaching NoSQL data modeling enhances students' ability to design scalable, application-specific databases, it also introduces challenges, such as understanding the trade-offs between flexibility and complexity in schema-less systems.

Query languages. Query languages in NoSQL courses is essential for equipping students with the skills to interact with diverse NoSQL systems, each with its unique approach to querying data. Understanding query languages such as MongoDB's query language, Cassandra Query Language (CQL), and Cypher for Neo4j enables students to perform CRUD operations on various NoSQL databases while working with data formats like JSON and XML (Amparo and Huang, 2024). Dong and Wang (2016) provided hands-on applications using Cassandra and MongoDB for practical scenarios like a student club data store and a high school social network, helping students grasp fundamental query operations. Mohan (2018) expanded this approach with structured labs on CRUD operations, integrating complex use cases such as multi-author book systems, check-in/check-out features, and recommendation engines, emphasizing the importance of justifying data models and query design choices. To simplify the learning curve, Alawini et al. (2022) developed methods for converting user-generated queries into SQL, Cypher, and MongoDB queries, bridging the gap for learners with limited database programming experience. Alkhabaz et al. (2021) highlighted the practical use of MongoDB's JavaScript-enabled shell, fostering familiarity with native query environments. For graph databases, Chen et al. (2021) showcased Neo4j's Cypher language, demonstrating its declarative structure and real-world applicability.

Scalability and performance optimization. Understanding scalability and performance optimization in NoSQL courses is vital for equipping students to manage big data effectively. Teaching scalability requires addressing their complexities through theoretical and practical methods. Lectures as highlighted in Kim (2020) could introduce core concepts such as horizontal scalability, and the role of sharding to support horizontal scalability and optimize both read and write performance. Hands-on labs enable students to deploy sharded clusters, troubleshoot issues, and analyze performance trade-offs (Mohan, 2018; Mitri, 2023). Mohan (2018) incorporated structured labs to teach scalability, comparing NoSQL and SQL databases while emphasizing trade-offs between scalability and consistency. Their approach included hands-on exercises like sharding and replication, allowing students to understand distributed data partitioning and replication for optimized. Using platforms like cloud based services or virtual machines, students can manage distributed datasets and address real-world challenges such as load balancing and maintaining consistency. Assignments and projects involving tools like MongoDB and Cassandra provide opportunities to explore query-driven data modeling and the performance impact of sharding (Silva et al., 2016; Sattar et al., 2013). Dong and Wang (2016) focused on teaching with instructional computing platforms in which NoSQL is aligned with cloud computing and large-scale data processing, helping students grasp the practical applications of scalability in NoSQL systems. Furthermore, Guo et al. (2016) explored the scalability of MongoDB, Cassandra, and Redis, focusing on their ability to handle big, fast, and diverse data in non-relational contexts.

Big data management. Teaching big data management in NoSQL courses is vital for equipping students to manage large-scale data collections using NoSQL databases (Mingyu et al., 2017). Grillenberger and Romeike (2014) emphasized the value of incorporating real-world big data examples and publicly available datasets into education, enabling students to develop hands-on skills and understand the statistical relevance critical for big data analysis. Poulová et al. (2016) highlighted the growing industry demand for professionals skilled in big data technologies, stressing the need to align curricula with roles like big data architects and analysts. Silva et al. (2016) advocated for integrating modern database technologies, such as MapReduce and NoSQL to provide a holistic framework for big data education. This approach ensures students gain experience in managing and processing large datasets across multiple systems. While this prepares students for diverse applications, the challenge lies in the complexity of mastering multiple tools and frameworks within a single course (Grillenberger and Romeike, 2014).

Security. Incorporating security into NoSQL database courses is essential for equipping students with the skills to recognize and address the unique security risks inherent to these systems. Guo et al. (2016) and Li et al. (2016) both outlined a comprehensive approach to NoSQL security education. Guo et al. (2016) developed a series of hands-on labs for MongoDB that addressed key security topics, including authentication, authorization through Role-Based Access Control (RBAC), and data encryption. These labs included activities on detecting and mitigating NoSQL injection attacks, securing data at rest, and defending against connection pollution. This comprehensive design enabled students to

Table 8
Learning Environments Mentioned in the Sources.

Aspect	Source
Face to face, in campus	Chen et al. (2021), Dong and Wang (2016), Fowler et al. (2016), Guo et al. (2016), Wu (2023)
Hybrid/blended	Mitri (2023), Zinovieva et al. (2021), Werner (2022)
Online, distance learning	Li et al. (2016), Mingyu et al. (2017), Poulová et al. (2016), Werner and Bach (2018), Werner (2022), Zinovieva et al. (2021), Ehlers (2024)

explore security threats and solutions both practically and conceptually. Li et al. (2016) introduced the REALAB (Role-Based Learning Labware), a framework combining role-based learning with hands-on labs. Students alternated between attacker and defender roles (war mode) to simulate real-world cybersecurity scenarios. For example, in the NoSQL injection lab, students practiced attacking MongoDB systems and implemented defense mechanisms such as parameterized statements and input validation. The benefit of these hands-on, role-based approaches is that they offer immersive, experiential learning opportunities that encourage critical thinking and problem-solving.

Answer to RQ2. Based on the above observations, the most taught NoSQL models in higher education settings include document and graph databases. Document-oriented databases, such as MongoDB, are particularly popular, offering flexibility in handling semistructured data. Graph databases, such as Neo4j, are taught for their ability to model and query complex, interconnected data, with students learning query languages, such as Cypher. The topics covered often include core NoSQL concepts, data modeling, query languages, performance optimization, and scaling techniques. In particular, NoSQL education emphasizes the growing need for handling big data and their applications, integrating cloud platforms for practical learning. Security concerns, such as NoSQL injection and authorization, are also taught to prepare students for real-world challenges. Tools such as MongoDB, Neo4j, and Redis are frequently used in coursework to provide hands-on learning experiences. These tools are introduced through practical projects, allowing students to engage with real-world data and understand the strengths and limitations of each database model.

6. Teaching approaches

Teaching in NoSQL education has evolved to accommodate the dynamic and diverse nature of NoSQL technologies. Given the practical and rapidly changing landscape of higher education, educators are adopting innovative learning environments and teaching methods to ensure that learners not only grasp the theoretical underpinnings of and gain hands-on experience with these systems but also enjoy flexibility in terms of their locations and preferences. Some effective teaching approaches used in NoSQL education are discussed below.

6.1. Learning environment

Higher education learning environments have evolved to include several modalities: traditional Face-to-face (F2F) learning, fully online or e-learning, distance learning, and flexible hybrid/blended learning, as seen in primary sources (see Table 8). Face-to-face learning offers direct, in-person interaction, while online and distance learning provide flexibility for students to learn remotely, often at their own pace. Hybrid and blended models combine in-person and online elements, balancing flexibility with personal interaction.

Face-to-face or contact teaching. Face-to-face learning is critical in fostering direct interaction, offering immediate support from instructors, and allowing for hands-on learning experiences with active learning that can enhance students' technical skills and knowledge retention (Ananga and Biney, 2017). F2F teaching plays a vital role in NoSQL courses, particularly for mastering data models and complex technical concepts. F2F teaching environments combine preparatory work, interactive lectures, and hands-on sessions, fostering active learning and real-time problem-solving. Dong and Wang (2016) highlighted the benefits of F2F learning by having students set up and manage Amazon EC2 instances as part of their coursework, allowing them to gain practical experience under direct instructor supervision with immediate feedback. Similarly, Guo et al. (2016) emphasized the importance of in-person collaboration in mobile and NoSQL database security labs, where students actively engage with instructors and peers to troubleshoot and implement security measures. However, this approach can be resource-intensive and may not accommodate all digital learning platforms (Stevens et al., 2021). Despite these limitations, F2F teaching provides unparalleled opportunities for active engagement and practical learning, making it an indispensable component of NoSQL education.

Online, distance, or e-learning. Online and distance learning play a pivotal role in NoSQL education by offering flexible, scalable, and accessible learning opportunities tailored to diverse student needs. Effective strategies, such as structured video lectures, reading materials, and online labs, ensure that learners can acquire theoretical knowledge and apply it in practical contexts (Mitri, 2023). Platforms like PrairieLearn, as discussed by Alkhabaz et al. (2021), enhance the online learning experience by enabling students to submit MongoDB queries and receive instant feedback via auto-graders. Advanced labs, such as mobile database security exercises developed by Li et al. (2016), empower students to engage in hands-on activities from anywhere, highlighting the flexibility of distance education. MOOCs, explored in Mingyu et al. (2017), further expand accessibility by incorporating diverse database techniques into self-paced modules, making NoSQL education scalable for global audiences.

Authors in Werner and Bach (2018), Werner (2022) emphasized the value of modular online content tailored to industry needs, promoting lifelong learning and addressing skill gaps through flexible schedules. To enhance engagement, learning management systems, online programming judge systems, and collaborative tools such as discussion forums, virtual hackathons, and peer reviews foster interaction and teamwork (Zinovieva et al., 2021). Gamification elements, such as challenges and leaderboards, engage participants by showcasing the top five individuals with the most correct solutions. This approach creates a competitive yet enjoyable learning experience (Ehlers, 2024). Despite its advantages, online learning may lack the immediacy of face-to-face interaction and requires careful design to maintain student motivation (Stevens et al., 2021).

Hybrid or blended learning. Hybrid or blended teaching is an innovative approach to effectively delivering NoSQL database education by integrating traditional face-to-face (F2F) instruction with online learning components. This method leverages the strengths of F2F interaction, such as real-time feedback, in-class discussions, and hands-on problem-solving, while also taking advantage of the flexibility provided by online resources and assignments (Raes, 2022). Zinovieva et al. (2021) emphasize that communication between instructors and students is crucial in hybrid environments, as it ensures clarity of expectations and fosters engagement. Werner (2022) and Mitri (2023) explored the adjustments made during the COVID-19 pandemic, during which hybrid models became essential for teaching technical subjects like NoSQL databases. Their studies outline strategies to maintain educational effectiveness, including restructuring course schedules, enhancing online content, and integrating collaborative tools. The 50/50 blended learning model discussed by Zinovieva et al. (2021) combines 50% synchronous instruction—alternating between in-person and live virtual sessions—with 50% asynchronous content delivery through

Table 9
Learning outcomes discussed in the sources.

Aspect (Forehand, 2010)	Description
Remembering	In terms of learning outcomes related to remembering NoSQL concepts, students are expected to retrieve, recognize, and recall relevant knowledge from long-term memory. For example, in Alawini et al. (2022), the learning outcomes for students were to understand and recall data entities and their connections, demonstrating the retrieval of knowledge about NoSQL database structures. Similarly, the students in Li et al. (2016) were expected to recall fundamental concepts of SQL injection, including its attack vectors and potential risks. Learners were also tasked with recalling basic storage structures and access techniques pertinent to NoSQL systems, which form the foundation for understanding advanced database management principles (Mohan, 2018).
Understanding	With regard to understanding NoSQL concepts, students are expected to construct meaning from oral, written, and graphic messages by developing the ability to comprehend and contrast different data models and query languages. This involves constructing an understanding of SQL, NoSQL, and MapReduce, enabling learners to grasp their distinct features and applications (Wu, 2021). Additionally, students were tasked in Mohan (2018) with identifying and explaining the semantic similarities and differences between new and known databases, which helps them understand in-depth insights into database systems.
Applying	In terms of applying NoSQL concepts, students are expected to carry out or use procedures in practical situations by leveraging their theoretical knowledge. For example, in Hwang (2018), students applied their understanding by creating and modifying a graph database using SQL and processing it with database built-in algorithms. Additionally, hands-on experience with cloud computing and industry-grade tools in Dong and Wang (2016) enabled students to apply their learning in real-world data processing scenarios.
Analyzing	With regard to analyzing NoSQL concepts, students are expected to break down complex material into its constituent parts and discern the relationships between these parts and the overall structure. For instance, when analyzing database injection security risks, students decompose the concepts into vulnerabilities, attack vectors, and corresponding defense strategies, gaining a clear understanding of how these elements interact and contribute to database security (Li et al., 2016). Additionally, the students in Mohan (2018) identified the semantic similarities and differences between databases and differentiated between various consistency models, which helped them detect relationships within and across database paradigms.
Creating	In terms of creating, students are expected to put elements together to form a coherent or functional whole by reorganizing or transforming various components into new patterns or structures. For instance, students demonstrate creativity by transforming JSON-based schemas into different database structures, showcasing their ability to adapt and restructure data formats (Alawini et al., 2022). They also build user-centric, cloud-based database applications, integrating various technologies and user needs into a unified system (Wu, 2021). Furthermore, students develop information systems that utilize polyglot persistence, applying multiple database technologies to create complex, scalable systems (Mohan, 2018).

Table 10
Teaching Methods Used To Teach NoSQL.

Aspect	Source
Project-based learning	Dong and Wang (2016), Fowler et al. (2016), Kim (2020), Mitri (2023), Mohan (2018), Poulová et al. (2016), Sattar et al. (2013), Wang and Wang (2023), Wu (2021)
Hands-on Labs	Alawini et al. (2022), Dong and Wang (2016), Guo et al. (2016), Li et al. (2016), Mohan (2018), Wu (2023), Ehlers (2024), Mitri (2023), Kotsifakos et al. (2019)
Interactive Lectures	Kim (2020), Krasheninnik et al. (2022), Mohan (2018), Poulová et al. (2016), Zinovieva et al. (2021), Ehlers (2024)
Research Seminars	Kim (2020), Mohan (2018)

platforms like Moodle. This balanced structure helps mitigate common challenges of purely online education, such as student isolation and uneven workload distribution.

6.2. Learning outcomes and teaching methods

At the center of education are learning outcomes, which represent the key knowledge, skills, and abilities that students are expected to showcase following the conclusion of a course or program. These outcomes are typically categorized into remembering, understanding, applying, analyzing, and creating, as mentioned in Forehand (2010). We evaluated the learning outcomes of a NoSQL education source based on these aspects, and the results are summarized in Table 9.

The teaching methods in NoSQL education have evolved to accommodate the dynamic and diverse nature of NoSQL technologies. The following are some effective teaching methods used in NoSQL education as noted in primary sources and shown in Table 10.

Project-based learning. Project-based learning (PBL) is a highly effective approach in NoSQL courses, fostering the application of theoretical knowledge to real-world scenarios while enhancing problem-solving skills. Studies such as Dong and Wang (2016), Fowler et al. (2016), and Kim (2020) demonstrate the benefits of PBL in enabling students to develop comprehensive projects, such as building NoSQL-based applications or conducting performance comparisons between systems. For instance, Fowler et al. (2016) illustrated how NoSQL concepts could be seamlessly integrated into traditional database management courses through simple, practical projects. Kim (2020) detailed a PBL approach where students engaged in end-to-end application development using MongoDB, involving tasks like use case analysis, data modeling, and system implementation, effectively bridging theoretical and practical learning.

Moreover, PBL promotes understanding of system requirements and tailored solutions. Sattar et al. (2013) introduced hybrid system projects to teach the integration of relational and NoSQL databases for solving business problems. Similarly, Mohan (2018) enabled students to work on client-oriented projects, exposing them to real-world expectations and applications. Wu (2021) incorporated cloud-hosted NoSQL databases into PBL, where students transformed relational data into NoSQL formats and developed apps for data exploration. While PBL enhances hands-on experience and contextual learning, it can be time-intensive and requires careful guidance to ensure project goals are met. Nonetheless, PBL equips students with critical technical and analytical skills, making it an indispensable strategy for teaching NoSQL in modern educational contexts.

Hands-on labs. Hands-on labs in NoSQL database courses can improve students' practical understanding of NoSQL security and various querying mechanisms across different NoSQL models (Li et al., 2016). Alawini et al. (2022) demonstrated the value of hands-on labs by introducing TriQL, which helps students explore generalized schemas and query languages across relational, graph, and document databases. Similarly, Dong and Wang (2016) incorporated cloud infrastructure into labs, comparing platforms like Amazon EC2 and Hood VCL, with students favoring EC2 for its ease of use in data processing exercises. Guo et al. (2016) and Li et al. (2016) focused on security labs, using real-world scenarios and open-source tools to simulate ethical hacking and teach defenses against security vulnerabilities in mobile and NoSQL databases. Mohan (2018) designed labs on polyglot persistence, requiring students to develop a library information system using multiple NoSQL databases, fostering an understanding of feature distribution across diverse systems. Hands-on labs enhance problem-solving skills, reinforce practical NoSQL knowledge, and prepare students for real-world challenges, though they can be resource-intensive and require careful alignment with industry needs.

Interactive lectures. Interactive lectures are vital in NoSQL courses as they combine theoretical instruction with active engagement, fostering a comprehensive understanding of NoSQL technologies and their applications. Kim (2020) demonstrated the effectiveness of transitioning seamlessly from fundamental concepts to practical knowledge, using hands-on assignments in MongoDB's data modeling and operations to reinforce lecture content. Mini-lectures, as discussed by Krasheninnik et al. (2022), focused on specific topics like database types and selection criteria, providing targeted insights into solving real-world problems. Mohan (2018) emphasized active learning through interactive sessions, incorporating live programming, data modeling exercises, scalability discussions, and quizzes. These methods engage students directly with complex NoSQL concepts while promoting critical thinking and collaboration. Interactive lectures can enhance knowledge retention, provide immediate feedback, and enable application of theory in a collaborative setting, but require careful planning to balance content and participation for effective NoSQL education.

Research seminar. Integrating research seminars into NoSQL courses is essential for fostering advanced understanding and developing students' research and critical thinking skills. Kim (2020) highlighted how research seminars enhance students' interest in NoSQL technologies by encouraging them to explore cutting-edge developments and theoretical concepts through the study and presentation of recent research papers. This approach promotes deep engagement with emerging trends and fosters a nuanced understanding of the NoSQL domain. Mohan (2018) emphasized the value of replicating real-world experiences, where students research and present unique NoSQL databases, equipping them with the ability to adapt to new technologies and evaluate them based on project requirements. Research paper presentations, as reported by Kim (2020), encourage peer learning, critical evaluation, and collaboration through group discussions, Q&A sessions, and feedback exchanges. Research seminars enhance analytical skills and industry readiness, though they require substantial preparation and can challenge students new to academic research, ultimately fostering a deeper understanding of NoSQL technologies for both academic and professional contexts.

6.3. Learning assessment

The integration of innovative teaching methodologies and learning assessment strategies in database education is essential for addressing the diverse and evolving needs of students. Several studies have highlighted various approaches to enhancing learning outcomes and assessing student performance effectively.

Online platforms reflect the growing recognition of the need for assessments that can effectively measure students' understanding of complex database concepts. Alawini et al. (2022) discussed the integration of online assessment tools, such as PrairieLearn, and the support for various data models in their work. The authors in Chen et al. (2021), Alkhabaz et al. (2021) highlighted the importance of online assessment platforms in database education, pointing to their use in evaluating student solutions to MongoDB and graph database problems. **Peer correction** is an approach to learning assessment that emphasizes the value of peer feedback and collaboration in the learning process. Wu (2023) focused on assessing peer correction of SQL and NoSQL queries, finding that students can achieve high accuracy rates over different styles of assessment questions and that peer correction helps gain new perspectives on writing queries.

Assignments in NoSQL database courses play a vital role in helping students bridge the gap between theoretical knowledge and practical application. They offer a nuanced approach to learning modern database management. By working on specific assignments that utilize Neo4j and CouchDB, students are challenged to apply complex queries and operations, fostering a hands-on comprehension of various NoSQL systems (Chen et al., 2021), Fowler et al. (2016). These tasks are designed not only to reinforce practical skills but also to ensure that students can effectively navigate the intricacies of NoSQL databases. This is evidenced by exercises on MongoDB and Cassandra (Kim, 2020). The use of formal tools in assignments further underscores the importance of a solid conceptual foundation, enabling students to validate their understanding through practical application (Howard, 2015), Hwang (2018). Instructors have implemented various innovative **examination** strategies to effectively assess student learning and understanding. Kim (2020) varied their exam formats between the open-book and closed-book techniques across different years to reflect real-world conditions. Wu (2021) designed their final exams to evaluate individual modules' effectiveness in their course by offering a comprehensive range of questions. Another innovative approach was reported in Wu (2023), and it involved designing multiple-choice questions based on common midterm mistakes to assess students' capabilities in identifying and correcting these mistakes. This approach directly addressed misunderstandings and fostered in-depth learning.

6.4. Teaching assessment

Teaching assessments in NoSQL database courses can provide valuable insights into students' learning experiences, outcomes, and perceptions. Pre-course surveys (Dong and Wang, 2016; Fowler et al., 2016), post-course surveys (Dong and Wang, 2016; Kim, 2020; Fowler et al., 2016), along with continuous anonymous surveys (Mohan, 2018), are essential tools for evaluating and improving course effectiveness. In one NoSQL course, the authors conducted pre- and post-surveys to assess students' database skills before and after the course implementation (Dong and Wang, 2016). The results from these surveys can offer educators feedback on their teaching methods, helping them achieve desired learning outcomes and enhance students' critical technological competencies. In a different study on NoSQL education (Mohan, 2018), continuous anonymous surveys were used to gather ongoing feedback from students. These continuous surveys enable educators to make real-time adjustments to the course, fostering a sense of ownership among students and effectively addressing their expectations and learning needs.

Answer to RQ3. In higher education, NoSQL databases are taught using various pedagogical approaches that blend theoretical knowledge with practical application. Project-based learning allows students to work on real-world projects, applying NoSQL technologies to solve industry-relevant problems. Hands-on labs provide practical experience with tools, such as MongoDB, Cassandra, and Neo4j, enhancing students' technical skills. Interactive lectures combine theoretical lessons with in-class exercises and discussions to deepen understanding. Hybrid or blended learning models offer flexibility by mixing F2F and online learning. Additionally, research seminars encourage students to explore advanced NoSQL topics, fostering peer learning and engagement. These methods ensure a comprehensive, hands-on educational experience in NoSQL database management.

6.5. NoSQL education challenges

For the fourth RQ, some challenges were identified in the primary articles regarding teaching NoSQL databases, spanning various aspects of curriculum design, pedagogical strategies, and practical implementation. These challenges can be categorized in Table 11.

Answer to RQ4. The key challenges in teaching NoSQL databases involve integrating diverse NoSQL technologies, addressing varied

Table 11
Challenges in NoSQL education.

Challenge	Description
Diverse student backgrounds (Fowler et al., 2016), Wu (2021)	NoSQL Courses are often tasked to students with varied academic and professional backgrounds, making it essential to adopt a flexible and inclusive approach to course design and delivery. Fowler (Fowler et al., 2016) emphasized the challenge of accommodating students with different levels of prior experience when introducing NoSQL databases. Similarly, Wu (2021) highlighted the need for adaptive teaching strategies that addressed the distinct learning needs of students, from those new to NoSQL systems to those with substantial technical expertise.
Practical skills development in distance learning (Grillenberger and Romeike, 2014), Zinovieva et al. (2021)	One of the key challenges in distance learning for NoSQL database courses is developing practical skills without direct access to physical NoSQL database servers and real-world database interactions. Grillenberger (Grillenberger and Romeike, 2014) discussed how simulating real-world scenarios, such as managing large datasets in big data contexts, is essential but difficult to achieve remotely. Similarly, Zinovieva (Zinovieva et al., 2021) highlighted the hurdles in replicating hands-on experiences in virtual settings, in which students may lack the resources or infrastructure to engage with complex NoSQL systems.
Adaptation to online and blended learning models (Werner and Bach, 2018), Zinovieva et al. (2021)	The shift to online and blended learning models necessitates a thorough reevaluation of traditional teaching methodologies to maintain effectiveness in these new environments. Werner (Werner and Bach, 2018) emphasized the need to redesign course content for asynchronous and synchronous online formats, ensuring that NoSQL core concepts are effectively communicated through interactive materials and remote lab exercises. Similarly, Zinovieva (Zinovieva et al., 2021) highlighted the importance of integrating flexible learning tools to teach NoSQL, such as virtual labs and cloud-based platforms, to replicate hands-on experiences and maintain student engagement in blended learning models.
Teaching through translation and comparison (Wu, 2021)	Developing effective teaching materials that help students understand and apply knowledge across different database paradigms is a significant challenge. Wu (Wu, 2021) emphasized the importance of creating resources that not only introduce students to the technical aspects of both relational and NoSQL databases but also guide them in translating concepts between these paradigms.
Use of MOOCs and other online resources (Zinovieva et al., 2021)	Incorporating MOOCs and other online resources into formal education as supplementary materials necessitate careful planning and evaluation to ensure that they align with course objectives and enhance learning. Zinovieva (Zinovieva et al., 2021) highlighted the challenges of selecting appropriate MOOCs, which must not only complement the curriculum but also provide students with the flexibility to explore NoSQL topics at their own pace.
Ensuring accessibility and inclusivity (Werner, 2022)	Addressing technical and pedagogical barriers is critical to ensuring that all students have equal access to learning materials and can fully participate in the NoSQL course. Werner (Werner, 2022) emphasized the need for inclusive design in online and hybrid learning environments, in which students may face challenges related to internet access, device compatibility, or varying levels of technical proficiency.

student backgrounds, and adapting to online and blended learning environments. Integrating NoSQL technologies into a coherent curriculum is challenging due to the diversity of NoSQL data models and query languages. Another challenge is accommodating students with diverse academic backgrounds, which requires adaptive teaching strategies to ensure effective engagement. Practical NoSQL skills development is also difficult in distance learning, where access to real-world infrastructure is limited, making virtual labs and simulations necessary. Additionally, adapting content for online formats involves redesigning NoSQL course materials to maintain effectiveness. Teaching through translation between relational and NoSQL paradigms requires effective comparative resources while incorporating MOOCs and ensuring accessibility demand careful alignment with NoSQL course objectives and inclusivity efforts.

7. Discussion

This section covers NoSQL education, teaching methods, recommendations for educators, and the validity of the study.

7.1. NoSQL education and teaching methods

Our goal here is to compare various teaching approaches related to NoSQL topics, providing insights into the outcomes presented in the Results section.

Learning environments versus teaching method. Teaching methods, such as PBL, hands-on labs, interactive lectures, and seminars as seen in Table 10, are employed to foster practical skills and theoretical understanding. These methods are adapted to fit within diverse learning environments—F2F, online, and hybrid models (Table 8). Each learning environment poses unique challenges. For instance, online and hybrid models demand innovative strategies to ensure engagement and effective learning, leveraging technology to bridge physical distance. Conversely, F2F environments offer direct interaction but must also evolve to incorporate digital tools and resources that prepare students for the technological demands of their future careers. Table 12 shows the alignment of teaching methods with the chosen learning environment to maximize educational effectiveness.

NoSQL educators can effectively adapt teaching methods to fit diverse learning environments by leveraging the unique strengths and addressing the challenges of face-to-face, online, and hybrid models. F2F approaches enhance engagement through live coding, data modeling exercises, and quizzes (Kim, 2020; Mohan, 2018), while hands-on labs provide practical skills in NoSQL configuration and security, though constrained by physical resources (Dong and Wang, 2016; Guo et al., 2016). Research seminars and collaborative projects in F2F settings foster immediate feedback and critical understanding of NoSQL technologies (Fowler et al., 2016; Mohan, 2018). In online environments, educators can leverage virtual labs with platforms like MongoDB Atlas, interactive query simulations, and live project-based learning sessions via tools like Zoom and Miro to ensure engagement and participation (Ehlers, 2024; Poulová et al., 2016). Hybrid models blend synchronous online lectures with asynchronous lab assignments and in-person sessions for troubleshooting and collaborative work, offering flexibility while maximizing learning outcomes (Guo et al., 2016; Kim, 2020). By integrating adaptive learning systems, gamified exercises, and real-world projects across these environments, educators can bridge gaps in engagement, equip students with practical skills, and prepare them for the demands of NoSQL technologies.

Learning outcomes versus teaching method. In NoSQL database education, aligning teaching methods (Table 10) with learning outcomes (Table 9) ensures that students develop key cognitive abilities. Interactive lectures and hands-on labs effectively support remembering by offering structured recall opportunities, as noted in Alawini et al. (2022) and Kim (2020). For understanding, labs provide practical comparisons, while lectures offer theoretical insights, as described in Wu (2021) and Mohan (2018). Applying knowledge is best facilitated through PBL and labs, allowing students to work on real-world tasks and focused exercises, as outlined in Dong and Wang (2016) and Mohan (2018). Analyzing is enhanced through research projects and interactive lectures that promote deep, independent analysis and real-time problem solving, as mentioned in Kim (2020) and Li et al. (2016). Finally, creating is encouraged through PBL and labs, enabling large-scale system development and fostering creativity, as emphasized in Fowler et al. (2016) and Mohan (2018). By matching teaching

Table 12
Learning environments versus the use of teaching methods.

Aspect	Interactive lectures	Hands-on labs	Project-based learning	Research seminar
Face to Face	While traditionally effective through direct interaction, integrating interactive elements like live coding, data modeling exercises, and quizzes into lectures enhances student participation and engagement (Kim, 2020), (Mohan, 2018).	Hands-on labs offer high engagement and effectiveness by providing practical skills in areas like security testing and NoSQL database configuration, though they are constrained by physical space and resource availability (Dong and Wang, 2016; Wu, 2023; Guo et al., 2016; Li et al., 2016).	This approach fosters collaboration and immediate feedback through comprehensive projects, though it offers less flexibility (Dong and Wang, 2016) (Fowler et al., 2016).	Traditional F2F settings offer strong mentorship and peer interaction, with research seminars fostering critical discussions and feedback to deepen understanding of NoSQL technologies (Mohan, 2018).
Online	Online platforms offer high adaptability with multimedia and interactive tools, enabling real-time discussions, polls, and quizzes to enhance engagement, though they may lack interaction depth (Zinovieva et al., 2021; Ehlers, 2024; Krasheninnik et al., 2022; Poulová et al., 2016).	Remote labs using platforms like AWS, Google Cloud, and Azure provide adaptable and engaging opportunities for database setup, querying, and performance testing (Ehlers, 2024; Dong and Wang, 2016; Li et al., 2016).	Projects leveraging digital tools and collaborative platforms offer flexibility and effectiveness, enabling students to develop real-world applications and solve complex problems online (Wu, 2021; Poulová et al., 2016).	Online systems offer high adaptability and access to extensive digital resources, enabling students to research topics like NoSQL databases and present findings via video conferencing tools, though engagement may vary (Kim, 2020).
Hybrid	Combining live and recorded sessions promotes flexibility, effectiveness, and engagement, as interactive face-to-face lectures with demonstrations are enhanced by online quizzes and assignments (Zinovieva et al., 2021; Kim, 2020).	A hybrid approach combining online pre-lab lectures with in-person lab sessions maximizes learning outcomes by blending hands-on activities with remote access to industry-standard tools (Guo et al., 2016; Li et al., 2016).	This approach combines online resources and F2F support, enabling students to tackle real-world problems collaboratively while balancing flexibility with guided, self-paced learning (Mitri, 2023; Kim, 2020).	This facilitates deep research through a mix of online resources and F2F mentorship, enhancing both engagement and effectiveness.

methods to specific learning outcomes, students not only acquire theoretical knowledge but also gain practical and analytical skills crucial for success in NoSQL database technologies.

Teaching methods versus NoSQL topic. NoSQL education encompasses key topics, such as core concepts, data modeling, query languages, scaling, big data, and security (see Section 5.3). These topics are effectively taught using various methods, such as PBL, hands-on labs, interactive lectures, and research projects (Table 10). For core concepts, interactive lectures are essential for providing foundational knowledge, and they are supported by hands-on labs for practical reinforcement (Mohan, 2018; Wang and Wang, 2023). Data modeling benefits from PBL for real-world problem solving and hands-on labs for schema design and querying practice (Kim, 2020; Alawini et al., 2022). In teaching query languages, hands-on labs enable mastery of CRUD operations, while interactive lectures explain syntax and logic (Dong and Wang, 2016), Alawini et al. (2022). Scaling and performance optimization are best taught through PBL, in which students apply scalability concepts in real-world scenarios and hands-on labs for step-by-step performance optimization (Alkhabaz et al., 2021; Guo et al., 2016). Big data are introduced through interactive lectures on integration with NoSQL, while PBL allows students to build big data systems (Silva et al., 2016; Wu, 2021). NoSQL security is taught using hands-on labs for practical exposure to security risks, with research projects offering an in-depth exploration of security challenges (Guo et al., 2016; Kim, 2020). By aligning teaching methods with these NoSQL topics, educators can create a comprehensive learning experience that balances theory with practical skills.

7.2. Recommendations for educators

Online learning platforms. Educators are encouraged to adopt digital platforms to expand access to education, enhance the learning experience with a variety of multimedia resources, and support self-paced learning for students (Liu et al., 2020). These platforms also provide the flexibility to accommodate diverse learning styles and paces, which is particularly advantageous in the fast-evolving field of database technology. For example, NoSQL educators can improve online learning by incorporating interactive tutorials on platforms like MongoDB University or DataCamp, allowing for hands-on experience with NoSQL databases (Zinovieva et al., 2021). They can design modular learning paths within Learning Management Systems (LMS), covering foundational concepts such as document-oriented databases, indexing, and scalability, followed by advanced topics like sharding and replication. Educators can utilize multimedia resources—such as video demonstrations of querying and schema design—and assign collaborative projects (Werner, 2022; Zinovieva et al., 2021). Additionally, tracking student progress through analytics can further reinforce knowledge and skill application.

Use of cloud-based NoSQL services. It is recommended that educators incorporate practical experiences with cloud-based NoSQL databases as noted in Fowler et al. (2016) into their teaching strategies to provide students with hands-on, real-world experience. This approach helps students grasp the scalability, flexibility, and performance advantages of NoSQL databases in cloud environments, aligning with current industry trends and demands. Educators can create hands-on labs where students set up, query, and optimize NoSQL databases in the cloud. This highlights important concepts such as scalability, distributed systems, and data partitioning (Mohan, 2018). This not only enhances students' learning experience by offering exposure to cutting-edge technologies but also prepares them for the challenges and opportunities in the tech industry. Assignments can include tasks like deploying cloud-based NoSQL solutions, performing big data analytics, and implementing replication and sharding strategies. By leveraging free-tier, educators can provide cost-effective access to these technologies, preparing students to meet industry demands while fostering a deep understanding of modern data management challenges (Dong and Wang, 2016; Silva et al., 2016).

Use of artificial intelligence (AI) in NoSQL education. Databases are essential in various emerging technologies, including mobile applications and artificial intelligence (Zinovieva et al., 2021; Zhou et al., 2020). The integration of modern technologies like AI into NoSQL databases has been discussed (Asaad, 2023), and incorporating these technologies into curricula enhances learning by demonstrating the practical applications of database systems in real-world scenarios. For instance, a chatbot-based learning platform to help students learn SQL has been reported in Balderas et al. (2024). The paper (Nalintippayawong et al., 2017) introduces an integrated architectural approach system that serves as a personalized and adaptive e-learning platform specifically designed to assist students taking database courses. It emphasizes essential yet challenging topics for newcomers, including database design and SQL commands. Integrating AI into NoSQL education can significantly enhance learning by offering personalized, adaptive, and interactive support for complex subjects. AI-driven chatbots can serve as virtual teaching assistants, answering questions and guiding students through challenges such as schema design and query syntax. Adaptive learning platforms can identify students' weaker areas and recommend tailored exercises, while generative AI can create realistic datasets and lab scenarios for hands-on practice. Additionally, AI tools can provide instant feedback on assignments.

Use of graphical UIs in NoSQL education. Graphical UIs used by educators can offer their courses intuitive and user-friendly interfaces that enable students to visualize database structures, execute queries easily, and receive immediate feedback, outperforming command-line or DB-specific UIs. For instance, Alawini et al. (2022) introduce TriQL, a tool to learn schema and query languages of MySQL, Neo4J, and MongoDB through a graphical interactive user interface without programming experience, making it ideal for labs and assignments. Similarly, Ehlers (2024) presents a unified tool for relational, document-oriented, and graph-based models, where students solve queries, receive instant feedback, and leverage gamification to enhance learning. Additionally, article (Meyer et al., 2024) describes a digital tool to teach NoSQL databases, allowing students to directly execute queries and focus on mastering query languages while benefiting from features like learning analytics.

7.3. Validity discussion and study limitation

In the context of NoSQL education and SLR, the discussion of our study's validity is based on criteria from previous studies. Following these criteria—external validity, conclusion validity, internal validity, and construct validity—we have identified potential threats to our study's validity and implemented strategies to mitigate them, as recommended by Zhou et al. (2016).

External validity. External validity refers to the extent to which the findings of a study can be generalized beyond the specific context of the research. To enhance the generalizability of our study, we ensured that the keywords related to NoSQL and education were consistent with those used in prior systematic reviews on the topic. We also selected primary articles that discussed both pedagogy methods and NoSQL topics in higher education settings. This ensures that the data extracted from these studies are relevant and robust in addressing the RQs. A potential threat to external validity is the restricted time frame of the reviewed literature. To mitigate this, we did not impose any time restrictions on our database searches, capturing literature up until 2024.

Conclusion validity. Conclusion validity concerns the appropriateness of the methods used to derive the conclusions and whether similar outcomes would be obtained upon repeating the study. This study was conducted by a researcher experienced in SLRs and data analysis, ensuring a rigorous and methodical approach to the search, data extraction, and synthesis processes. A potential threat to conclusion validity is the replication of studies across different sources. To address this, we employed the Parsifal tool to automate aspects of the literature review process, including the identification and removal of duplicate articles.

Construct validity. Construct validity involves ensuring that the theoretical concepts investigated are adequately represented in the research and that these align with the study's objectives. Our RQs are explicitly designed to address our research aims, which focus on exploring the current state of research on NoSQL education, its teaching methods, and the specific NoSQL topics covered in educational settings. A potential threat to construct validity could arise from ambiguities in the description of the systematic review process. To address this, we developed a review protocol within the SLR tool, adhering to established guidelines from previous literature, which helped ensure clarity and consistency in the identification of primary studies. The use of the Parsifal tool also contributed to specifying and documenting the review settings, further strengthening construct validity.

Study limitation. The study's reliance on a single author presents both advantages and limitations. The author's extensive experience in teaching NoSQL courses and conducting literature reviews (Tripathi et al., 2016, 2024) offers valuable practical insights and a solid foundation for exploring this field. However, the lack of multi-author collaboration may result in potential drawbacks, such as an absence of diverse perspectives that could enhance the study's comprehensiveness. While the analysis of 32 primary studies is systematic, as shown in Table 3, involving multiple researchers could help reduce biases, enhance objectivity, and provide more nuanced interpretations of the findings. Furthermore, while the literature review methodology is thorough, it raises questions about its completeness in reflecting the current state of NoSQL education. The study utilizes reputable databases and incorporates snowballing to capture additional sources, but it may exclude relevant gray literature, such as white papers and web materials on NoSQL education, which often contain practical insights.

8. Conclusion

Integrating NoSQL databases into educational curricula is increasingly important, as big data and cloud-based infrastructures reshape modern data management. Traditional SQL-based courses, while still essential, do not fully address the complexities and scalability demands of real-world applications. Nonstructured query language databases offer flexibility in data modeling, scalability, and the ability to handle unstructured data, making them vital for students entering industries that rely heavily on big data. Despite this, there is a noticeable gap in academic research on the most effective teaching methodologies and practices for NoSQL education. This study's SLR seeks to bridge such a gap by exploring current approaches to teaching NoSQL databases, including data modeling, scaling, and management strategies, ultimately contributing to the development of a standardized NoSQL curriculum that aligns with industry requirements.

This research also emphasizes the importance of evidence-based teaching practices for NoSQL education, focusing on widely used models and databases, such as MongoDB, Cassandra, and Neo4j. It highlights the need for hands-on experience in areas such as scalability and security while addressing the pedagogical challenges educators face, such as adapting to diverse student backgrounds and incorporating online learning platforms. By providing knowledge on NoSQL topics in education and use of teaching approaches for teaching NoSQL, this study aims to equip educators with the tools necessary to update course content and keep pace with rapidly evolving technologies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Ahmed, M. Razu, Khatun, M. Arifa, Ali, M. Asraf, Sundaraj, Kenneth, 2018. A literature review on NoSQL database for big data processing. Int. J. Eng. Technol. 7 (2), 902–906.

Alawini, Abdussalam, Rao, Peilin, Zhou, Leyao, Kang, Lujia, Ho, Ping-Che, 2022. Teaching data models with TriQL. In: 1st International Workshop on Data Systems Education. pp. 16–21.

Alkhabaz, Ridha, Poulsen, Seth, Chen, Mei, Alawini, Abdussalam, 2021. Insights from student solutions to mongodb homework problems. In: Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education, vol. 1, pp. 276–282.

Amparo, Xavier, Huang, Ching-Yu, 2024. Mastering NoSQL through SQL foundations. In: 2024 IEEE Integrated STEM Education Conference. ISEC, IEEE, pp. 01–05.

Ananga, Patricia, Biney, Isaac Kofi, 2017. Comparing face-to-face and online teaching and learning in higher education. MIER J. Educ. Stud. Trends Pr. 165–179.

Asaad, Chaimae, 2023. Towards leveraging artificial intelligence for NoSQL data modeling, querying and quality characterization. In: 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE, pp. 192–198.

Askarbekuly, Nursultan, Solovyov, Alexandr, Lukyanchikova, Elena, Pimenov, Denis, Mazzara, Manuel, 2021. Building an educational product: Constructive alignment and requirements engineering. In: Advances in Artificial Intelligence, Software and Systems Engineering: Proceedings of the AHFE 2021 Virtual Conferences on Human Factors in Software and Systems Engineering, Artificial Intelligence and Social Computing, and Energy. July 25–29, 2021, USA, Springer, pp. 358–365.

Balderas, Antonio, Baena-Pérez, Rubén, Person, Tatiana, Mota, José Miguel, Ruiz-Rube, Iván, 2024. Chatbot-based learning platform for SQL training.

Biggs, John, Tang, Catherine, Kennedy, Gregor, 2022. Ebook: Teaching for Quality Learning at University 5e. McGraw-hill education (UK).

Bjeladinovic, Srdja, Marjanovic, Zoran, Babarogic, Sladjan, 2020. A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components. J. Syst. Softw. 168, 110633.

Byrne, B., Nelson, David, Jayakumar, Renuga, 2017. Big data technology-can we abandon the teaching of normalisation? In: EDULEARN17 Proceedings. IATED, pp. 510–517.

Cain, Andrew, Babar, Muhammad Ali, 2016. Reflections on applying constructive alignment with formative feedback for teaching introductory programming and software architecture. In: Proceedings of the 38th International Conference on Software Engineering Companion. pp. 336–345.

Chen, Jeang-Kuo, Lee, Wei-Zhe, 2019. An introduction of NoSQL databases based on their categories and application industries. Algorithms 12 (5), 106.

Chen, Mei, Poulsen, Seth, Alkhabaz, Ridha, Alawini, Abdussalam, 2021. A quantitative analysis of student solutions to graph database problems. In: Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education, vol. 1, pp. 283–289.

Corbellini, Alejandro, Mateos, Cristian, Zunino, Alejandro, Godoy, Daniela, Schiaffino, Silvia, 2017. Persisting big-data: The NoSQL landscape. Inf. Syst. 63, 1–23.

Cruzes, Daniela S., Dyba, Tore, 2011. Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement. IEEE, pp. 275–284.

Dong, Aijuan, Wang, Baoying, 2016. Exploring instructional computing platforms for a small liberal arts college: A case study with nosql. In: 2016 International Conference on Computational Science and Computational Intelligence. CSCI, IEEE, pp. 332–337.

Ehlers, Jens, 2024. Teaching multiple data models and query languages. In: Proceedings of the 2024 on Innovation and Technology in Computer Science Education, vol. 1, pp. 234–240.

Forehand, Mary, 2010. Bloom's taxonomy. Emerg. Perspect. Learn. Teach. Technol. 41 (4), 47–56.

Fowler, Brad, Godin, Joy, Geddy, Margaret, 2016. Teaching case: introduction to NoSQL in a traditional database course. J. Inf. Syst. Educ. 27 (2), 99.

German, Adrian, Salmeron, Santiago, Ha, Wonyong, Henderson, Bo, 2016. Mean web development: a tutorial for educators. In: Proceedings of the 17th Annual Conference on Information Technology Education. pp. 128–129.

Grillenberger, Andreas, Romeike, Ralf, 2014. Big data–challenges for computer science education. In: Informatics in Schools. Teaching and Learning Perspectives: 7th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2014, Istanbul, Turkey, September 22-25, 2014. Proceedings 7. Springer, pp. 29–40.

Guo, Minzhe, Qian, Kai, Yang, Li, 2016. Hands-on labs for learning mobile and NoSQL database security. In: 2016 IEEE 40th Annual Computer Software and Applications Conference. COMPSAC, 2, IEEE, pp. 606–607.

Han, Jing, Haihong, Ee, Le, Guan, Du, Jian, 2011. Survey on NoSQL database. In: 2011 6th International Conference on Pervasive Computing and Applications. IEEE, pp. 363–366.

Hiles, Mohanad, Agha, Massoud, 2017. Knowledge-based its for teaching mongo database.

Howard, Brian T., 2015. Leveraging synergy between database and programming language courses.

Hwang, Mark, 2018. Graph processing using SAP HANA: A teaching case. E-J. Bus. Educ. Sch. Teach. 12 (2), 155–165.

Hynninen, Timo, Kasurinen, Jussi, Knutas, Antti, Taipale, Ossi, 2018. Guidelines for software testing education objectives from industry practices with a constructive alignment approach. In: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education. pp. 278–283.

Khan, Wisal, Kumar, Teerath, Zhang, Cheng, Raj, Kislay, Roy, Arunabha M., Luo, Bin, 2023. SQL and NoSQL database software architecture performance analysis and assessments—A systematic literature review. Big Data Cogn. Comput. 7 (2), 97.

Kim, Suneuy, 2020. Seamless integration of nosql class into the database curriculum. In: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education. pp. 314–320.

Kitchenham, Barbara, Charters, Stuart, et al., 2007. Guidelines for performing systematic literature reviews in software engineering.

Kotsifakos, Dimitrios, Magetos, Dimitrios, Veletsos, Alexandros, Douligeris, Christos, 2019. Teaching the basic commands of nosql databases using Neo4j in vocational education and training (vet). Eur. J. Eng. Technol. Res. 13–18.

Krasheninnik, IV, Koniukhov, SL, Osadcha, KP, Chorna, AV, Serdiuk, IM, 2022. The impact of psychological and learning training on educational motives and reflective skills of future IT specialists. J. Phys. Conf. Ser. 2288 (1), 012039.

Li, Lei, Qian, Kai, Chen, Qian, Hasan, Ragib, Shao, Guifeng, 2016. Developing hands-on labware for emerging database security. In: Proceedings of the 17th Annual Conference on Information Technology Education. pp. 60–64.

Liu, Zi-Yu, Lomovtseva, Natalya, Korobeynikova, Elena, 2020. Online learning platforms: Reconstructing modern higher education. Int. J. Emerg. Technol. Learn. (IJET) 15 (13), 4–21.

Martinez-Mosquera, Diana, Navarrete, Rosa, Lujan-Mora, Sergio, 2020. Modeling and management big data in databases—A systematic literature review. Sustainability 12 (2), 634.

Menzin, Margaret, Mohan, Sriram, Musicant, David R, Sooriamurthi, Raja, 2020. Nosql in undergrad courses is noproblem. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education. pp. 962–963.

Meyer, Vanessa, Wiese, Lena, Al-Ghezi, Ahmed, 2024. A unified teaching platform for (no) SQL databases.

Mingyu, Lin, Jianping, Jun, Yi, Zhu, Cuili, Zhan, 2017. Research on the teaching reform of database curriculum major in computer in big data era. In: 2017 12th International Conference on Computer Science and Education. ICCSE, IEEE, pp. 570–573.

Mitri, Michel, 2023. Teaching case: Using python and AWS for NoSQL in a BI course. J. Inf. Syst. Educ. 34 (1), 41–48.

Mohan, Sriram, 2018. Teaching NoSQL databases to undergraduate students: A novel approach. In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. pp. 314–319.

Nalintippayawong, Srinual, Atchariyachanvanich, Kanokwan, Julavanich, Thanakrit, 2017. DBLearn: Adaptive e-learning for practical database course—An integrated architecture approach. In: 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. SNPD, IEEE, pp. 109–114.

Nayak, Ameya, Poriya, Anil, Poojary, Dikshay, 2013. Type of NOSQL databases and its comparison with relational databases. Int. J. Appl. Inf. Syst. 5 (4), 16–19.

Pasterk, Stefan, Bollin, Andreas, 2017. Graph-based analysis of computer science curricula for primary education. In: 2017 IEEE Frontiers in Education Conference. FIE, IEEE, pp. 1–9.

Poulová, Petra, Šimonová, Ivana, Borkovcová, Monika, El-Hmoudová, Dagmar, 2016. Big data EduCloud: New educational potential. DIVAI 2016.

Raes, Annelies, 2022. Exploring student and teacher experiences in hybrid learning environments: Does presence matter? Postdigital Sci. Educ. 4 (1), 138–159.

Sattar, Abdul, Lorenzen, Torben, Nallamaddi, Keerthi, 2013. Incorporating NoSQL into a database course. Acm Inroads 4 (2), 50–53.

Shreevatsan, SS, Kumar, Aniket, Hitesh, R, Kumar, Navin, Shailendra, Samar, 2024. SchemaVerse: Game-based learning of NoSQL queries. In: 2024 IEEE International Conference on Electronics, Computing and Communication Technologies. CONECCT, IEEE, pp. 1–6.

Silva, Yasin N., Almeida, Isadora, Queiroz, Michell, 2016. SQL: From traditional databases to big data. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. pp. 413–418.

Stanier, Clare, 2012. Introducing NoSQL into the database curriculum. In: 10th International Workshop on the Teaching, Learning and Assessment of Databases. p. 61.

Stevens, Garry John, Bienz, Tobias, Wali, Nidhi, Condie, Jenna, Schismenos, Spyros, 2021. Online university education is the new normal: but is face-to-face better? Interact. Technol. Smart Educ. 18 (3), 278–297.

Taipalus, Toni, 2019. A notation for planning SQL queries. J. Inf. Syst. Educ. 30 (3), 160–166.

Taipalus, Toni, Seppänen, Ville, 2020. SQL education: A systematic mapping study and future research agenda. ACM Trans. Comput. Educ. (TOCE) 20 (3), 1–33.

Tripathi, Nirnaya, Annanperä, Elina, Oivo, Markku, Liukkunen, Kari, 2016. Exploring processes in small software companies: a systematic review. In: Software Process Improvement and Capability Determination: 16th International Conference, SPICE 2016, Dublin, Ireland, June 9-10, 2016, Proceedings 16. Springer, pp. 150–165.

Tripathi, Nirnaya, Hietala, Heidi, Xu, Yueqiang, Liyanage, Reshani, 2024. Stakeholders collaborations, challenges and emerging concepts in digital twin ecosystems. Inf. Softw. Technol. 107424.

Vera-Olivera, Harley, Guo, Ruizhe, Huacarpuma, Ruben Cruz, Da Silva, Ana Paula Bernardi, Mariano, Ari Melo, Holanda, Maristela, 2021. Data modeling and nosql databases-a systematic mapping review. ACM Comput. Surv. 54 (6), 1–26.

Wang, Hai, Wang, Shouhong, 2023. Teaching tip: Teaching NoSQL databases in a database course for business students. J. Inf. Syst. Educ. 34 (1), 32–40.

Werner, Aleksandra, 2022. Applying NoSQL data adapter with the learning paths mechanism for better knowledge transfer in the age of distance learning. Procedia Comput. Sci. 207, 3330–3339.

Werner, Aleksandra, Bach, Małgorzata, 2018. NoSQL E-learning laboratory—Interactive querying of MongoDB and CouchDB and their conversion to a relational database. In: Man-Machine Interactions 5: 5th International Conference on Man-Machine Interactions. ICMMI 2017 Held At KrakÓw, Poland, October 3–6, 2017, Springer, pp. 581–592.

Wu, Wensheng, 2021. SQL2X: Learning SQL, NoSQL, and MapReduce via translation. In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education. pp. 590–596.

Wu, Wensheng, 2023. Assessing peer correction of SQL and NoSQL queries. In: Proceedings of the 54th ACM Technical Symposium on Computer Science Education, vol. 1, pp. 535–541.

Xiong, Wei, Hawley, Douglas, Monismith, David, 2015. NoSQL in database education: incorporating non-relational concepts into a relational database course: panel discussion. J. Comput. Sci. Coll. 30 (5), 151–152.

Zhou, Xuanhe, Chai, Chengliang, Li, Guoliang, Sun, Ji, 2020. Database meets artificial intelligence: A survey. IEEE Trans. Knowl. Data Eng. 34 (3), 1096–1116.

Zhou, Xin, Jin, Yuqin, Zhang, He, Li, Shanshan, Huang, Xin, 2016. A map of threats to validity of systematic literature reviews in software engineering. In: 2016 23rd Asia-Pacific Software Engineering Conference. APSEC, IEEE, pp. 153–160.

Zinovieva, I.S., Artemchuk, V.O., Iatsyshyn, Anna V., Romanenko, Y.O., Popov, O.O., Kovach, V.O., Taraduda, D.V., Iatsyshyn, Andrii V., 2021. The use of MOOCs as additional tools for teaching NoSQL in blended and distance learning mode. J. Phys. Conf. Ser. 1946 (1), 012011.