Journal of
Information
Systems
Education

Volume 36
Issue 1
Winter 2025

Teaching Tip
Integrating NoSQL Databases Into a Data Analytics Course

Madhav Sharma and Andy Bowman

Recommended Citation: Sharma, M., & Bowman, A. (2025). Teaching Tip: Integrating NoSQL Databases Into a Data Analytics Course. *Journal of Information Systems Education*, 36(1), 25-36. https://doi.org/10.62273/IQJV6614

Article Link: https://jise.org/Volume36/n1/JISE2025v36n1pp25-36.html

Received: July 5, 2024
First Decision: October 4, 2024
Accepted: January 2, 2025
Published: March 15, 2025

ISSN: 2574-3872 (Online) 1055-3096 (Print)

Journal of Information Systems Education, 36(1), 25-36, Winter 2025
https://doi.org/10.62273/IQJV6614

Teaching Tip
Integrating NoSQL Databases Into a Data Analytics Course

Madhav Sharma
College of Business
Kansas State University
Manhattan, KS 66506, USA
madhavsharma@ksu.edu

Andy Bowman
College of Business
East Carolina University
Greenville, NC 27858, USA
bowmanan22@ecu.edu

ABSTRACT

"Not only SQL" (NoSQL) databases have become widespread across organizations, enabling sophisticated, data-driven applications to be highly available, distributed, and cloud-based, such as e-commerce, social media, online multiplayer games, and video streaming. However, NoSQL is still sparsely found in MIS and analytics curricula. This teaching tip presents an experiential learning-based, five-module course structure for teaching analytics students structured query language (SQL) and NoSQL databases. We describe our implementation, where students learned relational databases and four types of NoSQL databases, with assessments conducted using use cases, projects, and exams. Students reported high levels of engagement and positive first-hand practice experiences with NoSQL beyond general concepts. We believe this course design will empower students to broaden their skill set and communicate more effectively about the Extract, Transform, Load (ETL) and Extract, Load, Transform (ELT) processes.

Keywords: NoSQL, Database, Data management, ETL, Data analytics

1. INTRODUCTION

Data analytics has become a central component of information systems curricula at universities. Some business schools have restructured their Management Information Systems (MIS) departments to place greater emphasis on this area (Urbaczewski & Keeling, 2019). Data engineering and the Extract, Transform, and Load (ETL) processes—also referred to as Extract, Load, and Transform (ELT) in some paradigms—are crucial foundational steps in data analytics (Boehler et al., 2020). Relational databases and structured query language (SQL) are staples of database syllabi (Wang & Wang, 2023). Along with the rapid development of big data analytics and cloud computing, more and more organizations are using "not only SQL" (NoSQL) databases in addition to traditional relational databases (Bajaj & Bick, 2020). The NoSQL database market has been projected to reach USD 36.46 billion by 2029, with a remarkable annual growth rate of 30% from 2023 to 2029 (Maximize Market Research, 2022).

Teaching NoSQL databases to typical information systems students lags far behind the development of NoSQL databases (Gessert et al., 2017; Wang & Wang, 2023). Even after decades of widespread adoption of NoSQL databases, the information systems community continues searching for effective pedagogical approaches to include NoSQL in traditional undergraduate database courses (Bajaj & Bick, 2020).

The ability to use NoSQL databases is a critical skill for business students (Wang & Wang, 2023). This knowledge allows them to understand NoSQL use cases and develop the skills to leverage NoSQL for problem-solving. The benefits of NoSQL include compatibility with cloud infrastructure, the ability to handle large datasets and analytics, exceptional scalability across distributed systems, and creating better data scientists for the highly competitive workforce (Gessert et al., 2017).

This teaching tip outlines a five-module course focused on experiential learning of NoSQL databases (Clem et al., 2014; Thouin & Hefley, 2024). It is designed for students without prior knowledge of SQL or data engineering who aspire to become data scientists and analytics experts. With the growing demand for NoSQL skills in the job market (Hartzel & Ozturk, 2024), it is essential to immerse students in the various use cases and environments of NoSQL. The course builds on Wang and Wang (2023), extending it over an entire semester to provide hands-on technical experience with diverse NoSQL use cases and environments. Additionally, this paper includes student feedback and data on their experiences with NoSQL.

2. BRIEF BACKGROUND

Lending and Vician (2012) describe a teaching tip as research that provides an innovative solution to a teaching need. The emphasis of the tip is on the improvement and how it is implemented (Lending & Vician, 2012). To that end, our teaching tip is based on a course design drawn from prior research and other curricula in practice (Fowler et al., 2016; Mitri, 2023; Wang & Wang, 2023). It incorporates various forms of NoSQL databases.

Relational databases, which rely on structured data models and Structured Query Language (SQL) for database interaction, are widely considered traditional data models due to their widespread adoption. Most database courses cover data models (entity relationships), normalization, and CRUD operations (create, read, update, delete) based on relational databases (Harrington, 2016). In contrast, non-relational NoSQL databases utilize alternate data models that may not be entirely ACID (Atomicity, Consistency, Isolation, and Durability) compliant, allowing organizations to store vast amounts of unstructured data efficiently and quickly without the redundancy typical of relational models. The primary advantage of NoSQL databases is their ability to swiftly process large data volumes, making them ideal for highly available, distributed, cloud-based applications such as e-commerce, social media, online multiplayer games, and video streaming applications (Gessert et al., 2017). These applications collect data for analytics, decision-making, and product improvement. However, a significant challenge of teaching NoSQL databases compared to traditional ones is their technical complexity, as they often require specialized programming skills that many business students lack.

The saying "When all you have is a hammer, every problem looks like a nail" highlights the importance of exposing aspiring data scientists to both NoSQL and SQL simultaneously. This dual exposure helps prevent them from limiting their problem-solving approaches. Introducing NoSQL as a comparative technology for students familiar with traditional databases can enhance their understanding by building on their existing knowledge. Learning the different environments of NoSQL increases the breadth of knowledge of tools and techniques and provides a deeper understanding of the ETL process. NoSQL is also a skill that can lead to better salaries for students. Even in data analytics roles, where their job does not include ETL functions, they can communicate more effectively with ETL teams.

While data engineering is included in most analytics curricula, it is sometimes underemphasized (Bajaj & Bick, 2020; Wang & Wang, 2023). While designing the course, we prioritized experiential learning by incorporating theoretical knowledge, use cases, and student-led projects (Clem et al., 2014). Although we focus on experiential learning and critical thinking, this teaching tip is not solely motivated by theory. Students at the university where this course was implemented had strongly demanded a data engineering course. Our advisory board also recommended enhancing the course's data engineering component. Following are some examples of excepts from students about the demand for this course:

- *"Normally, students can learn Analysis [in other classes] but we can't learn data engineering [in other classes]."*
- *"The course is a critical piece that I feel had been missing from the MSDA course path. I am very happy with the 'start to finish' pipeline nature of the assignments in this course. I feel better prepared to work with databases/data in my future profession."*

In a systematic analysis of job postings and published research, Nasir et al. (2020) identified NoSQL and the ability to work with big data as highly valued specialized knowledge and experience in the current job market. Prior research in database pedagogy has primarily focused on relational databases, with limited exploration of the specific needs involved in teaching NoSQL concepts. Fowler et al. (2016) introduced a teaching case using CouchDB, an open-source document-based NoSQL database. In this case, students created a database using social media data and built reports based on their findings. Similarly, Mitri (2023) proposed a teaching case where data analytics students completed a guided technical project utilizing AWS to work with DynamoDB, a document store database. This project also involved connecting DynamoDB to Python and Power BI. Both teaching cases highlighted the relative scarcity of NoSQL-related content available for instructors in MIS and data analytics courses.

While these case studies introduced novel changes to the curriculum, they were limited to single, small-to-medium assignments focused exclusively on document-based databases. Given the variety of NoSQL databases used in modern, highly available applications, there is a growing need for comprehensive coursework that addresses the diverse topics under the umbrella of NoSQL. Wang and Wang (2023) emphasized the need for further research in teaching NoSQL. They developed a robust module on NoSQL within a traditional relational database course but acknowledged the necessity for future studies to extend their initial work. Future work included exploring additional assessments, materials, and learning outcomes associated with NoSQL instruction. To address this need, we developed a five-module course structure that covers a broader range of NoSQL concepts and multiple databases, providing students with the skills necessary for modern data management. The course structure is designed to be accessible to all students, regardless of their prior technical experience.

This teaching tip builds on the need highlighted by Wang and Wang (2023) and incorporates elements from teaching cases (Fowler et al., 2016; Mitri, 2023), aiming to better equip students with the skills to use new tools while minimizing unnecessary technical complexity often associated with NoSQL courses. We conducted follow-up evaluations during and after the course to gather student feedback on their experiences and perceptions. To assess the effectiveness of the course structure, we employed both quantitative and qualitative analyses, identifying strengths and areas for potential improvement.

Our primary contribution is providing an innovative solution to a teaching need – specifically, integrating NoSQL into a data analytics course. Our focus is on developing an in-depth understanding of the ETL/ELT process so that data scientists can improve communication with their ETL/ELT departments. Additionally, they can conduct effective read operations.

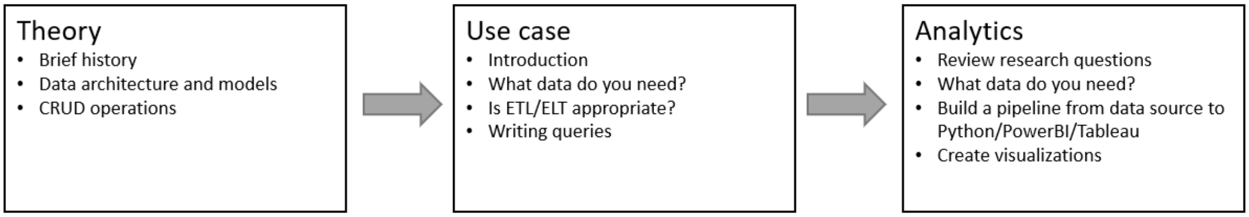
3. IMPLEMENTATION

3.1 Background of Database Course

Database management is an essential Management Information Systems (MIS) function (Boehler et al., 2020). In the university that implemented this course, database courses are open to all undergraduate business majors and are a core requirement for MIS students. However, the Master's in Data Analytics (MSDA) program did not include a dedicated database course. SQL and database knowledge were disseminated in other classes, such as Introduction to Data Analytics. Due to student demand, suggestions from the advisory board, and instructors' drive to promote data literacy, the university recognized the need to introduce a course that covers database concepts from the perspective of data analysts and data scientists. This course was open to all MSDA students and had no prerequisites. While the MIS course is geared towards file organizations, B-trees, designing relational models, entity relationship diagrams (ERD), normalization, and other topics related to relational database design techniques, the master's course for data analytics focused more on ETL/ELT techniques using databases. This entailed reading ERDs, writing complex queries to get the desired data in a desired form into an analytics terminal. Analytics techniques such as data visualization and predictive modeling are taught in other courses in this curriculum. We note that many universities already have such courses in place; our contribution is the course structure.

3.2 Five-Module Course Structure

The course was divided into five modules. Each module began with a theoretical portion covering the data architecture of each database, followed by a discussion of CRUD operations. Students were then introduced to a use case where they were given 15-20 problems requiring them to write queries that build a pipeline to retrieve specific data based on parameters. Each use case concluded with students connecting the database to an analytics terminal. Python scripts were provided to students, informing them about connection strings and authentication. Students were then asked to create data visualizations based on the data they retrieved from the pipeline. Appendix A shows the different modules and exercises. Each module is discussed in greater detail in the following sections. Each module followed the structure laid down in Figure 1. We have shared our detailed syllabus, course objectives, and their corresponding assessments to measure learning outcomes in the appendices.



Theory
- Brief history
- Data architecture and models
- CRUD operations

Use case
- Introduction
- What data do you need?
- Is ETL/ELT appropriate?
- Writing queries

Analytics
- Review research questions
- What data do you need?
- Build a pipeline from data source to Python/PowerBI/Tableau
- Create visualizations

Figure 1. Structure of Each Module

3.2.1 SQL and Relational Databases. SQL and relational databases constitute the first module. This module lays the foundation for the rest of the class. The module drew theoretical content from undergraduate MIS courses, starting with the three-tier architecture and explaining the importance of relational databases, including the principles of ACID compliance (Harrington, 2016). Students were introduced to entities, attributes, primary keys, and relationships, as well as how these relationships are enforced using foreign keys. Although the course was taught online, instructors recommend in-class exercises to teach foreign key constraints and reading ERDs in more detail, as there were many questions on these topics when covered.

CRUD operations using SQL were discussed with a focus on read operations. Create, update, and delete commands were demonstrated. Since the students are data analytics students and not MIS students, more emphasis was placed on the select command, various filters, temporary tables, and subqueries. A SQL proficiency assignment was provided with 20 questions covering simple queries, group by, join, and subqueries. Five questions were shown as references for the students. The assignment was reported as moderately challenging. Instructors can use any SQL model data to build these assignments.

A bonus assignment worth 10 extra points was also given for students further to demonstrate their SQL skills with a larger ERD. All but three students completed the assignment, which was reported to be of moderate to high difficulty. Foundational SQL skills are core to this course and indicate how well students will understand other concepts. The mid-term exam featured about 50% of its questions on SQL, ranging from objective questions about reading ERDs to query-based questions.

Overall, the instruction on SQL and relational databases took 2.5 weeks of the course. In the third week, we introduced the concept that not all applications run on relational databases. For highly available, distributed systems, NoSQL databases are used. The course provided a high-level explanation of each type of NoSQL database, along with the core concepts of ETL and ELT.

3.2.2 Document-Based Databases. Document-based databases offer distributed and resilient infrastructures, flexible schemas compared to relational databases, and object mapping (Araujo et al., 2021; Wang & Wang, 2023). We constructed this module and took structural and content cues from previous teaching cases on document-based databases. We chose MongoDB to teach the use case of a document-based database. MongoDB is widely used in the industry and has excellent teaching materials available. One of its key features is MongoDB Compass, a lightweight database graphical user interface (GUI) that allows students to visualize data and build queries. We also introduced the concept of containerization in this module, using Docker to enable students to avoid downloading multiple software applications. Docker can run many databases, but all students in the course were able to download and work with MongoDB Compass. Instructors note that there is also a MongoDB Atlas free tier that other instructors can leverage (https://www.mongodb.com/products/platform/atlas-database).

In the theoretical portion of this module, we covered how data is stored in BSON (similar to JSON) format in MongoDB.

Each record is a document, a group of documents is a collection, and multiple collections can be in a database. Concepts such as IDs, ObjectIDs, fields, field-value pairs, and relationships were discussed. Each of these concepts was mapped to their respective SQL relational database counterparts, and the instructors noted the differences. Theoretical portions of this and the following modules were drawn from various sources, including two textbooks (Harrison, 2015; Sullivan, 2015) and MongoDB university materials (https://learn.mongodb.com/).

Students were guided on how to download MongoDB Compass (https://www.mongodb.com/products/tools/compass). There was a compulsory check-in assignment where students had to submit screenshots to confirm they had installed the software. A use case based on publicly available data from Raleigh-Durham International Airport was provided (refer to Endnote 1). Students were required to run 10 simple and 10 aggregate MongoDB queries (refer to Endnote 2). They executed these queries in MongoDB Compass and submitted screenshots of each output.

Next, another use case was provided with publicly available financial services data. Students were given five queries and a research question. They were asked to build a data pipeline that extracts only relevant data into their analytics terminal (Python or Power BI). They were guided through this process. These queries were more complex than the previous ones and required unwind functions. Lastly, they were asked to generate a simple visualization based on the research question, with the option to create a more complicated visualization if they chose.

Overall, the MongoDB assignments were rated as moderately complex by the students. The data was intentionally kept small and manageable to ensure queries did not take too long. However, students were informed that queries would take longer with real-world datasets. Additionally, a bonus assignment was released, asking students to upload a dataset about movies to MongoDB and providing them with a list of research questions.

3.2.3 KV Stores. After an in-depth view of relational and document-based databases, a module on key-value (KV) stores was covered (Mitri, 2023; Wang & Wang, 2023). Redis (https://redis.io/) was chosen as the KV store for the use case. While many databases can function as KV stores, Redis is widely used and has a free cloud tier suitable for instruction called Redis Insight (https://redis.io/insight/). The theoretical portion of this module introduced KV stores and appropriate use cases, such as session data, caching, and shopping cart data, which are generally associated with KV stores. Key data structures were discussed in detail, including what can be a key and value and how values can be strings or containers (hashes, lists, sets, and sorted sets). Following the format of previous modules, CRUD operations were discussed.

A use case based on brewery data was selected. Although there are more suitable use cases for this data, this beer data was chosen to provide students with experience in different data types, as it contains tables with key-value pairs where values are stored in strings, hashes, and sorted lists. The data is publicly available online. The use case was primarily conducted in Python, with students asked to run about 10 queries. Due to the nature of the dataset and this database, an ETL operation—where a connection is established, a query run to select part of the data using CRUD and then loaded for final analytics—was deemed inappropriate. The network latency of the free tier and resources on the cloud are not suitable for this operation. Thus, an ELT approach was advised, where students pull most data from the Redis server and perform analytical operations themselves (Haryono et al., 2020). A few queries were demonstrated.

A few research questions were provided to help students build the ELT pipeline and generate visualizations. The assignment was rated as highly difficult due to slow network connections. Instructors advise using Docker (https://www.docker.com/) or an on-premises server if such a database is not available.

Due to university-level resource constraints and the nature of the database, this module was briefer compared to the previous ones. Data structure was discussed, and students and instructors identified which use cases are best suited for document-based databases, KV stores, and relational databases. Only the theoretical portion of KV stores was featured in the mid-term exam as objective questions.

3.3.4 Column Family Databases. A module on column family databases was covered (Araujo et al., 2021; Wang & Wang, 2023). It started with a brief history and an overview of significant databases that use column family architecture. It was noted that Google BigTable (https://cloud.google.com/bigtable) is one of the major players in the industry. Key features such as developer's dynamic control over columns, indexing using row identifiers, and fast atomic writes were highlighted as important considerations in column family databases.

As with the previous databases, concepts unique to column family databases and those common with other databases were discussed. The class used Apache Cassandra (https://cassandra.apache.org) as a use case. Cassandra specializes in distributed systems; hence, single Cassandra instances are called nodes, and many instances form a ring or cluster of nodes. They communicate with each other using a protocol called gossip. Students were also taught about partitions, data distribution, primary keys, keyspaces, and data centers. Demo queries were shown where the instructor created keyspaces with replication and tables with partition keys.

CRUD operations, specifically read operations, were emphasized. The CRUD commands in Apache Cassandra are very similar to those in SQL. As of March 2024, Apache Cassandra does not have Windows support. Therefore, Docker was used for containerization. This module also served as an informational module on containerization. The use case was brief. Students were shown how to upload data to a node using SQL and then connect it to a Python terminal to access and create a table.

Since this was a shorter use case and considering the CRUD similarities between SQL and Cassandra Query Language (CQL), a comprehensive SQL assignment was given. Students were asked to upload data to MSSQL/MySQL, connect tables, query them based on specific research questions, and generate visualizations.

3.3.5 Graph-Based Databases. Graph-based databases constituted the last instructional module (Besta et al., 2023; Kotiranta et al., 2022). In this module, students were familiarized with graph theory and its use cases. By now, due to exposure to four databases, students had developed a vocabulary and did not need much context. Some good graph scenarios were discussed, and the pros and cons of graph databases were demonstrated. Several graph-based databases were introduced; however, the use case was based on Neo4j (https://neo4j.com/). Many companies use Neo4j, which provides excellent learning opportunities. Students were introduced to components of graph data architecture, such as nodes, edges, labels, and properties.

CRUD operations using Neo4j's Cypher were discussed. Since graph data is hard to come by and difficult to set up, the preloaded datasets of Neo4j were used in this module. Students were asked to create an account in Neo4j's AuraDB (https://neo4j.com/cloud/platform/aura-graph-database/) cloud system. They selected preloaded data and wrote queries in the console with an emphasis on read operations. Lastly, as with all previous assignments, students were asked to connect to an analytics terminal using code and authentication information provided by Neo4j. They were then asked to create visualizations.

3.4 Major Projects

Aside from assignments and use cases for each module, students were given two significant unstructured projects during the class. Both projects had a 4-6-week lead time from release to submission. Once students got into the groove of the class after covering the first two databases, these projects provided a source of exploration and application for their own experience.

3.4.1 Exploration Project. The first unstructured project was an exploration project where students were asked to research a database not covered in class. Although we covered five types of databases, which is quite comprehensive, the NoSQL paradigm includes many different databases offered by various vendors and used by numerous companies. This assignment was provided after three out of the five databases had already been taught. Students were given ample time and were asked to create a report like how the instructor covered each database. The rubric is provided in the appendices. Most students produced detailed reports on Snowflake, examining CRUD operations (which are the same as SQL) and differences. They were asked to justify the use of their chosen database and explain why it was preferred over others, such as SQL relational databases, MongoDB, or Cassandra. These reports were a good way to test students' business analysis acumen as they rationalized the reasoning for using or not using each of the discussed databases.

3.4.2 Final Project. The second unstructured project was released with one month remaining in the class. This project was made optional (those who chose not to do it would receive a B) in case the effort expectancy was too overwhelming for students. These projects are beneficial for students' portfolios and allow them to apply the skills they have learned on a subject of their interest. Despite being optional, only 3 out of 29 students chose not to participate in the project. The project required students to choose two databases, install their data on them, build an ETL/ELT pipeline using queries, and answer research questions. Many interesting projects emerged from this assignment. Although most analytics classes focus on causality and models, this type of database assignment allowed students to explore topics that interested them. Some noteworthy projects included a spurious correlation analysis in which a student found interesting correlations, using electric car data on a Cassandra database and state population and gun store licenses in MongoDB databases. Similarly, another student project covered automobile recalls using MongoDB and MySQL databases. Overall, these projects allowed students to explore intriguing topics using their knowledge of multiple databases.

4. EVIDENCE AND DISCUSSION

Details of a recent classroom implementation are provided as evidence supporting the benefits of this course structure and content. This course structure was implemented in a graduate program at a large Midwestern university. The class included MS students from diverse backgrounds. Twenty-eight out of 29 students enrolled in the course completed all assignments and projects assigned. After completing the coursework, students were given assessments to provide feedback. One assessment was the Experiential Learning Survey (ELS), which was used to gauge the level of experiential learning (Clem et al., 2014; Thouin & Hefley, 2024). The other assessment was a survey about their prior knowledge of various databases and their confidence in their skills after the course. This survey also solicited qualitative feedback based on three questions: (1) What was your favorite part about this class? (2) What was your least favorite part about this class? (3) Any other burning comments/questions/suggestions? Both surveys were anonymous to ensure impartial feedback, and participants were awarded five points for completing them.

4.1 Experiential Learning Survey

In this survey, students evaluated their degree of agreement with statements from the ELS questionnaire on a 7-point Likert scale. The ELS questions pertain to four areas: authenticity, active learning, relevance, and utility (Clem et al., 2014). Higher scores represent higher levels of experiential learning. Table 1 summarizes the results.

As shown in Table 1, all four subscales were scored by students on the higher end of the possible ranges. Mean values for relevance were the highest (6.39), indicating that students found the course very relevant to their career goals. Similarly, mean scores for utility were also high (6.36), showing that students found the course very useful. Active learning mean scores also indicate that students felt engaged with the content throughout the semester. Authenticity mean scores were lower than the other subscales but aligned with other research in this area (Thouin & Hefley, 2024). Authenticity scores might have been impacted by the online medium of the class, limiting student interaction compared to an in-class environment. The question with the lowest score concerned interacting with people other than students and teachers, which is not entirely relevant to this content-based course as opposed to an outside project-based course like a capstone. Removing this question increases the average to 6.20, showing students had an authentic learning experience. This result is consistent with recent research on experiential learning (Thouin & Hefley, 2024). Overall, all subscales indicate a good experiential learning experience.

ELS Questions (7-point Likert Scale)	Mean Response (n=28)
Authenticity	
The way the course is set helps me understand the material better.	6.54
I expect real-world problems to come up during this learning experience.	6.11
The way the course is set does not enhance the learning experience.*	6.32
The learning experience requires me to interact with people other than students and teachers.	3.79
I expect to return to an environment similar to the one where this learning experience occurs.	5.82
Mean Authenticity	**5.71**
Active Learning	
I am stimulated by what I learned.	6.46
The learning experience requires me to do more than just listen.	6.14
The learning experience is presented to me in a challenging way.	5.79
I find this learning experience boring.*	6.17
I feel like I am an active part of the learning experience.	6.29
The learning experience requires me to really think about the information.	5.89
I am emotionally invested in this experience.	5.39
Mean Active Learning	**6.02**
Relevance	
I care about the information I am being taught.	6.54
The learning experience makes sense to me.	6.36
This learning experience has nothing to do with me.*	6.36
This learning experience is enjoyable to me.	6.32
I can identify with the learning experience.	6.32
This learning experience is applicable to me and my interests.	6.57
My educator encourages me to share my ideas and past experiences.	6.07
This learning experience falls in line with my interests.	6.54
I can think of tangible ways to put this learning experience into future practice.	6.43
Mean Relevance	**6.39**
Utility	
This learning experience will help me do my job better.	6.43
This learning experience will not be useful to me in the future.*	5.57
I will continue to use what I am being taught after this learning experience has ended.	6.21
I can see value in this learning experience.	6.71
I believe this learning experience has prepared me for other experiences.	6.54
I doubt I will ever use this learning experience again.*	6.42
I can see myself using this learning experience in the future.	6.64
Mean Utility	**6.36**
*Note: * Reverse Coded Item Corrected.*	

Table 1. Experiential Learning Survey (ELS) Results

4.2 Student Feedback

Students were asked to rate their experience and confidence in each core skill covered in the course using a set of questions that asked: "On a scale of 1-7, rate your overall skill level BEFORE/AFTER you began this course in the following areas." As shown in Table 2, students reported higher mean scores for their overall skills in each of these areas. The difference was highest in MongoDB, which is a document-based database. Mean scores for overall database concepts and SQL were higher on the "before" side, as these skills are commonly covered in undergraduate MIS and computer science curricula. Redis and Cassandra had lower mean scores compared to others on the "after" side. Although students gained suitable experience with these databases, due to resource constraints, their use could not be realized at a scale seen at the enterprise level.

We qualitatively analyzed students' answers to open-ended questions, and four major themes emerged (Braun & Clarke, 2006). Students noted the breadth of subjects, hands-on experience, theoretical knowledge, and technical challenges. Below, these themes are summarized.

Skill	Before (Mean)	After (Mean)
Concepts of Database	3.54	6.00
SQL	3.71	6.07
MongoDB	2.14	5.46
REDIS	1.79	4.71
Cassandra	2.07	4.54
Neo4j	2.04	4.89

Table 2. Self-Reported Improvement in Skills

4.2.1 Breadth of Subjects. Many students noted that their favorite part of the class was the number of subjects and technologies covered. This theme included responses like, "All the databases had something unique to offer," and "I was amazed that we covered 5 databases." These responses highlight the strength of using an entire class to cover a variety of technologies and ensuring exposure to them.

4.2.2 Hands-on Experience. Many students noted that they appreciated the hands-on experience and specific assignments in the course. This theme included responses like, "My favorite part was the hands-on demos for each assignment," and "I loved being able to be hands-on within the different database environments and develop my technical skills with the various querying exercises." These responses highlight that the students felt they gained value from the practical application of their knowledge and the guidance provided.

Some students noted that their favorite part of the course was using specific knowledge. While brief, the responses in this theme included comments like "*Utilizing Neo4j*." These responses show that despite covering multiple subjects within the course, some students found value in individual exposure.

4.2.3 Theoretical Knowledge. Some students identified that learning theory in the course was their favorite part. These responses included comments like "Learning about the engineering parts of the data." These responses demonstrate that some students prefer more abstract learning over specific technologies.

Other students identified theoretical knowledge as their least favorite aspect of the course. These responses included comments like "*Theory*" and "*I didn't really like the conceptual information on databases, but it is good information to know.*" Instructors note that the theoretical foundations of the database concepts are essential for meaningful experiential learning.

4.2.4 Technical Challenges With Software. Some students noted that their least favorite part of the course stemmed from technical limitations, either from their machines or the realities of data processing. Responses included, "Doing it on a Mac and having to take unique paths to get a software downloaded," and "Issues connecting to a database." These responses highlight the realities of working with technology in the classroom and demonstrate that no class will be without issues.

Other students identified specific assignments or technologies as their least favorite parts of the class. Responses included, "*I didn't like Redis or Cassandra as much*," and "*Working in Redis*." These responses help us understand the benefits of covering multiple technologies. Some students may have a negative experience with one technology, while others may enjoy it. By ensuring exposure to multiple technologies, students can leave the class with a sense of self-efficacy.

We also asked students to recommend improvements or make comments about the course. While students did respond, many only repeated comments and points made in the prior sections. Some requested technical improvements, others commented positively about the breadth of subjects covered, and some enjoyed the hands-on projects.

5. TEACHING SUGGESTIONS

Based on our implementation, we have the following suggestions for instructors who aim to implement our course design.

5.1 Relational Databases as a Starting Point

The current implementation was done in an MS in Data Analytics program. MS students have a variety of backgrounds. Due to this variety and college-level constraints, the course was offered with no prerequisites for database or SQL skills. While this created a time constraint, as earlier weeks of the course were used to teach relational databases, reiterating these concepts helped students understand NoSQL databases better. Thus, it is beneficial to start with a refresher on SQL and relational database concepts such as CRUD, ACID compliance, relationships, and foreign key constraints (Harrington, 2016). This foundation helps in the long run as new databases are introduced and students see the business need for deviating from relational databases for each of the new NoSQL databases.

For example, KV stores have faster write operations because they store data in a simple key-value pair format, allowing for quick data insertion and retrieval. This makes them ideal for applications that require high-speed data insertion, such as caching, session management, and real-time analytics (Chandramouli et al., 2018). On the other hand, column family databases, such as Apache Cassandra, offer advanced features like data center support modules, which enable seamless data replication and distribution across multiple data centers. This enhances data availability and fault tolerance, ensuring the system remains operational even during a data center failure (Araujo et al., 2021). Additionally, column family databases use gossip protocols for efficient communication between distributed nodes. Gossip protocols help nodes share state information about themselves and other nodes in the cluster, enabling the system to maintain consistency and coordinate data distribution effectively (Ben Brahim et al., 2016; Perez-Miguel et al., 2015).

By understanding the specific advantages and business needs that drive the use of each type of NoSQL database, students can better appreciate why certain technologies are chosen over traditional relational databases.

5.2 Use Cases and Projects

While our use cases and projects differed to provide a comprehensive database experience for students, we focused on read operations for these data science students. Extra emphasis in the wording of assignments and rubrics was on building an ETL/ELT pipeline that connects the database to their analytics tools (Python, Tableau, PowerBI) and generates visualizations. Students were explicitly asked to explain their steps in all cases.

Recurring questions for each assignment were: What data do you need? Why are you performing ETL or ELT in this case? In doing so, we highlighted the differences between ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform) processes, which are crucial for effective data integration and analytics. ETL is preferred when data requires extensive transformation before loading, making it ideal for legacy systems and batch processing. By leveraging modern data warehouses' power, ELT excels in scalability and real-time data processing by performing transformations within the target system (Haryono et al., 2020). Knowing when to use each approach helps optimize performance and align with specific business needs.

In most cases, the instructors provided two-three research questions, and students were asked to create one themselves. This ensured that they were thinking critically. Students were asked to provide a time breakdown at the end of each assignment. This had two main goals: first, it helped us understand the rigor required for each step. Instructors can sometimes overestimate or underestimate the difficulty level of assignments. Reporting time breakdowns ensured that we tailored the assignments properly. Second, it provided a real-world simulation of the analytical report development process. Consulting companies and data science departments often require employees to log time for project scheduling and budgeting purposes.

The instructor team benefited from creating new assignments based on publicly available datasets, which allowed us to present complex problems rooted in real-world and current settings. For instance, the data used in MongoDB was Raleigh-Durham Airport (RDU) flight data. Although this is dummy data, it is rich and exposes students to multiple functions. Similarly, beer data was used in Redis, basketball data in Cassandra, and store data in Neo4j. While other databases provide students with flexibility in understanding data structures, graph-based databases require data to be set in a specific way. Therefore, preloaded graph-based datasets were used in the Neo4j module.

5.3 New Tools and Techniques

Much of the instruction relied on free-tier software due to resource constraints. MSSQL, MySQL, MongoDB, Redis, and Neo4j offer great training platforms that instructors can leverage for teaching purposes. These databases provide a unique opportunity to build apps and host data for data analytics. Unfortunately, they pose many challenges, as students have different computers and may encounter compatibility issues. While most apps are compatible with Windows and Mac, Docker is a valuable resource that can help fill the gap using containerization. It is also an important concept to teach in technical courses.

This class's content is challenging and easier to manage with students' use of generative AI. Tools like Copilot (https://copilot.microsoft.com/) and ChatGPT (https://chatgpt.com/) enabled students to write complex queries and debug them. They were also able to understand more conceptual topics instead of getting bogged down by syntax errors. However, in some cases, students over-relied on ChatGPT, which produced incorrect answers. This was a learning experience in assessing the outputs of generative AI. As these tools proliferate in the industry, it is the instructors' task to integrate them into course design. The AI policy for this class was declarative: students were instructed to use AI and declare for what they used it. It was clear that generative AI should not be cited as a source of knowledge but as a tool for debugging or aggregating information.

5.4 Assessing Objective and Subjective Knowledge

Each assignment and project allowed students to flex their knowledge. In addition, multiple-choice exams were given to assess their objective knowledge (Lending et al., 2019; White et al., 2008). Students performed well in both exams, with the average score being a high B. While projects and use cases provide suitable learning environments where students can ask for help, exams test their objective conceptual knowledge for each database.

5.5 Challenges and Strategies to Overcome Them

Implementing the course design was not without its challenges. During execution, we faced three primary issues. The first challenge was resource constraints. This course was implemented at a large Midwestern university, where students used their own machines for all assignments and use cases. Working with four different databases introduced compatibility issues, which we addressed using Docker, a containerization system that allowed students to run all the databases as developer tools. Docker facilitated interaction with the databases through Python scripts and Azure Data Studio. Additionally, the product documentation for each database was invaluable in designing use cases, with resources such as free-tier GUIs and supporting slides from MongoDB, Redis, and Neo4j proving especially helpful.

Second, implementing this course with large class sizes can be challenging and require additional resources, such as teaching assistants, to provide adequate technical support. Ensuring sufficient help for students was essential to address technical difficulties effectively, especially given the complexities of working with multiple databases.

Third, we faced the challenge of balancing the breadth and depth of learning. Introducing students to four types of databases made it difficult to ensure they developed deep expertise in any one of them beyond the pre-written use cases. To address this, we incorporated exploration projects where students selected a database to study in more depth and justified why it was the best fit for their project. While these projects encouraged deeper learning, they also posed challenges in evaluation and were perceived as overly rigorous by some students. To mitigate this, we made the project optional. Students who completed all other coursework could receive a grade one level lower than those who participated in the project. Ultimately, only 10% of students opted out, indicating that most valued the additional learning opportunity.

6. CONCLUSION AND FUTURE RESERACH

Database courses play a critical role in developing the core knowledge and skills essential for data analysts and data scientists (Wang & Wang, 2023). Many use cases start with flat files and CSVs, but an extensive array of enterprise data is housed in databases. SQL and relational databases are staples of database courses in MIS curricula across universities. NoSQL databases have also become quite ubiquitous in curricula (Bajaj & Bick, 2020; Wang & Wang, 2023). Data analysts and scientists benefit from courses geared towards data management by becoming more effective in their jobs, communicating better with data engineering teams for their ETL/ELT needs, and enhancing their overall job performance. Labor market surveys consistently list SQL and database skills as high in demand. We implemented a five-module course structure that covered data architecture in various databases. Our assessments showed that students learned new skills and believed their skills improved in each database category and were engaged throughout the learning experience. We encourage future research to build on our modules and introduce alternative forms of highly available databases, such as Cassandra and Google BigTable, in the classroom. Additionally, some university curricula have enhanced access to cloud resources. We urge researchers to explore how data engineering modules such as ours can be implemented with AWS and Azure to present students with greater variety and a more authentic learning experience through real world use cases.

The presented class design relies heavily on resources and student engagement. The instructor team faced resource constraints when covering proprietary tools and technologies. Free tools were used. However, the licensing and availability of these tools are not permanent. Future instructors might face paywalls and loss of support for the free tiers of many NoSQL applications used. On the other hand, some alternative tools with free tiers might be more suitable. Resource-rich universities with robust IT support should consider providing specialized VMs with preloaded software for students to use. The instructors also note that they were fortunate to have an enthusiastic group of highly driven students. In their experience, they had not encountered a more motivated cohort willing to learn so many different tools in such a limited time. Other instructors should exercise their judgment when adapting to this structure.

As data-driven skills become increasingly in demand and generative AI pushes the boundaries of what students can do and what we can teach, integrating NoSQL into database courses for data analysts is a great idea. In this teaching tip, we presented a five-module course design that was very effective in our implementation. We hope instructors across universities will consider adopting this framework and improving it.

7. ENDNOTES

1. The authors are willing to provide detailed slides and assignments upon request. However, the assignments are not publicly shared or hosted online to preserve academic integrity.
2. All data and corresponding Python notebooks can be found at: https://github.com/kansasprofessor/Data-for-NoSQL-courses

8. REFERENCES

Araujo, J. M. A., de Moura, A. C. E., da Silva, S. L. B., Holanda, M., de Oliveira Ribeiro, E., & da Silva, G. L. (2021). Comparative Performance Analysis of NoSQL Cassandra and MongoDB Databases. *The 16th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-6). https://doi.org/10.23919/CISTI52073.2021.9476319

Bajaj, A., & Bick, W. (2020). The Rise of NoSQL Systems: Research and Pedagogy. *Journal of Database Management*, 31(3), 67-82. https://doi.org/10.4018/JDM.2020070104

Ben Brahim, M., Drira, W., Filali, F., & Hamdi, N. (2016). Spatial Data Extension for Cassandra NoSQL Database. *Journal of Big Data*, 3(1), article 11. https://doi.org/10.1186/s40537-016-0045-4

Besta, M., Gerstenberger, R., Peter, E., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., & Hoefler, T. (2023). Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. *ACM Computing Surveys*, 56(2), 1-40. https://doi.org/10.1145/3604932

Boehler, J. A., Larson, B., & Shehane, R. F. (2020). Evaluation of Information Systems Curricula. *Journal of Information Systems Education*, 31(3), 232-243.

Braun, V., & Clarke, V. (2006). Using Thematic Analysis in Psychology. *Qualitative Research in Psychology*, 3(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Chandramouli, B., Prasaad, G., Kossmann, D., Levandoski, J., Hunter, J., & Barnett, M. (2018). FASTER: Faster: A Concurrent Key-Value Store With In-Place Updates. *Proceedings of the 2018 International Conference on Management of Data* (pp. 275-290). https://doi.org/10.1145/3183713.3196898

Clem, J. M., Mennicke, A. M., & Beasley, C. (2014). Development and Validation of the Experiential Learning Survey. *Journal of Social Work Education*, 50(3), 490-506. https://doi.org/10.1080/10437797.2014.917900

Haryono, E. M., Fahmi, A. S., Tri W, I., Gunawan, A., Nizar Hidayanto, A., & Rahardja, U. (2020). Comparison of the E-LT vs ETL Method in Data Warehouse Implementation: A Qualitative Study. *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, (pp. 115-120). https://doi.org/10.1109/ICIMCIS51567.2020.9354284

Fowler, B., Godin, J., & Geddy, M. (2016). Teaching Case: Introduction to NoSQL in a Traditional Database Course. *Journal of Information Systems Education*, 27(2), 99-104.

Gessert, F., Wingerath, W., Friedrich, S., & Ritter, N. (2017). NoSQL Database Systems: A Survey and Decision Guidance. *Computer Science - Research and Development*, 32, 353-365. https://doi.org/10.1007/s00450-016-0334-3

Harrington, J. L. (2016). *Relational Database Design and Implementation*. Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-804399-8.00006-5

Harrison, G. (2015). Next Generation Databases: NoSQL and Big Data. Apress. https://doi.org/10.1007/978-1-4842-1329-2

Hartzel, K.S., & Ozturk, P., (2024). Tools for Success: Their Impact on Salaries in the Data Analytics Job Market. *Journal of Information Systems Applied Research*, 17(2), 45-60, https://doi.org/10.62273/JPTA5240

Kotiranta, P., Junkkari, M., & Nummenmaa, J. (2022). Performance of Graph and Relational Databases in Complex Queries. *Applied Sciences*, 12(13), 6490. https://doi.org/10.3390/app12136490

Lending, D., Mitri, M., & Dillon, T. W. (2019). Invited Paper: Ingredients of a High-Quality Information Systems

Program in a Changing IS Landscape. *Journal of Information Systems Education*, 30(4), 266-286.

Lending, D., & Vician, C. (2012). Writing IS Teaching Tips: Guidelines for JISE Submission. *Journal of Information Systems Education*, 23(1), 11-18.

Maximize Market Research. (2022). *NoSQL Database Market: Global Industry Analysis and Forecast (2023-2029)*. Maximize Market Research. https://www.maximizemarketresearch.com/market-report/global-nosql-database-market/97851/

Mitri, M. (2023). Teaching Case: Using Python and AWS for NoSQL in a BI Course. *Journal of Information Systems Education*, 34(1), 41-48.

Nasir, M., Dag, A., Young, W. A., & Delen, D. (2020). Determining Optimal Skillsets for Business Managers Based on Local and Global Job Markets: A Text Analytics Approach. *Decision Sciences Journal of Innovative Education*, 18(3), 374-408. https://doi.org/10.1111/dsji.12212

Perez-Miguel, C., Mendiburu, A., & Miguel-Alonso, J. (2015). Modeling the Availability of Cassandra. *Journal of Parallel and Distributed Computing*, 86, 29-44. https://doi.org/10.1016/j.jpdc.2015.08.001

Sullivan, D. (2015). *NoSQL for Mere Mortals*. Addison-Wesley Professional.

Thouin, M. F., & Hefley, W. E. (2024). Teaching Tip: Teaching Scrum Product Owner Competencies Using an Experiential Learning Simulation. *Journal of Information Systems Education*, 35(1), 37-47. https://doi.org/10.62273/GXMA1727

Urbaczewski, A., & Keeling, K. B. (2019). The Transition From MIS Departments to Analytics Departments. *Journal of Information Systems Education*, 30(4), 303-310.

Wang, H., & Wang, S. (2023). Teaching Tip: Teaching NoSQL Databases in a Database Course for Business Students. *Journal of Information Systems Education*, 34(1), 32-40.

White, B., Longenecker, H., McKell, L., & Harris, A. L. (2008). Assessment: Placing the Emphasis on Learning in Information Systems Programs and Classes. *Journal of Information Systems Education*, 19(2), 165-168.

AUTHOR BIOGRAPHIES

Madhav Sharma is an assistant professor of management information systems at Kansas State University. He completed his Ph.D. in Management Science and Information Systems from Oklahoma State University. His current research deals with human-AI interactions, future of work, and deception detection. His work has been published in *AIS Transactions on Human-Computer Interactions*, *Journal of Information Systems Education*, and many other journals and conference proceedings. He likes to party in Ibiza.

Andy Bowman is an assistant professor of management information systems at East Carolina University. He completed his Ph.D. in Management Science and Information Systems from Oklahoma State University. His current research deals with digital piracy, human-AI interactions, and privacy. His work has been published in *Drake Management Review*, *Journal of the Midwest Association for Information Systems*, and many conference proceedings. He enjoys watching movies and playing video games.

APPENDICES

Appendix A. Course Structure

Module	Topic	Assignments
Module 1	Syllabus and Course Intro	
	Introduction to Databases (SQL/NoSQL)	
	Relational Databases and SQL Primer	
	Queries in SQL/MySQL	
	Document Based Databases	
Module 2	MongoDB install	SQL Proficiency Assignment
	MongoDB use case	
	MongoDB use case	
	MongoDB use case	SQL Proficiency Bonus
	MongoDB use case	
Module 3	Key Value-Based Databases	MongoDB Use Case 1
	Redis install	
	Redis use case	MongoDB Use Case 2
	Redis use case	
	Redis use case	MongoDB Proficiency Bonus
	Mid Term Exam	
	Spring Break	
Module 4	Column-Based NoSQL Databases	
	Cassandra install	
	Cassandra use case	Redis Use Case
	Cassandra use case	
Module 5	Graph Based NoSQL Databases	
	Neo4j install	
	Neo4j use case/Project Check ins	NotCQL Assignment/ Exploration Project
	Neo4j use case/ Project Check ins	
	Neo4j use case/ Project Check ins	
	Neo4j use case/ Project Check ins	
	Neo4j use case	
	Neo4j use case	
Misc Topics	Evaluation and Optimization	Neo4j Use case
	Trends and New Directions	
	Final Exam	
	Final Exam	Class Project

Appendix B. Course Objectives and Corresponding Assessments to Measure Learning Outcomes

Course Objectives (Students should be able to…)	Corresponding Assessment
Understand the underlying principles that differentiate NoSQL from traditional RDBMS, discerning their unique strengths and appropriate use cases.	Use Cases Mid Term Exam
Analyze the trade-offs involved in choosing one NoSQL database over another, considering factors like consistency, availability, and partition tolerance.	Mid Term Exam Exploration Project Class Project
Solve complex, real-world problems through the application of appropriate strategies and the use of logical reasoning skills.	Use Cases Class Project
Connect to databases. Create and populate relational and NoSQL database.	Use Cases
Demonstrate competency in selecting a particular NoSQL database for specific use cases.	Class Project
Demonstrate the ability to construct queries tailored to different NoSQL databases, acknowledging the unique syntax and considerations of each.	Use Cases Mid Term Exam Class Project
Competently perform ETL operations as data analysts.	Use Cases Class Project

INFORMATION SYSTEMS & COMPUTING ACADEMIC PROFESSIONALS

iscap

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the ***Journal of Information Systems Education*** have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN: 2574-3872 (Online) 1055-3096 (Print)