Teaching Case
Introduction to NoSQL in a Traditional Database Course

Brad Fowler
Joy Godin
Margaret Geddy
Department of Information Systems and Computer Science
Georgia College & State University
Milledgeville, GA 30161, USA
brad.fowler@gcsu.edu, joy.godin@gcsu.edu, margaret.geddy@gcsu.edu

ABSTRACT

Many organizations are dealing with the increasing demands of big data, so they are turning to NoSQL databases as their preferred system for handling the unique problems of capturing and storing massive amounts of data. Therefore, it is likely that employees in all sizes of organizations will encounter NoSQL databases. Thus, to be more job-ready, college students need to be introduced to this technology to begin to have a functional understanding of how it works and how to use it. This paper provides a simple project-based, teaching case that introduces NoSQL and can be easily integrated into any existing database management course to augment concepts and skills geared around traditional SQL relational databases. The teaching case was tested and student feedback (pre- and post-assessment results, shown in the data analytics and results section) indicated a significant increase in their basic knowledge of NoSQL.

Keywords: NoSQL, Database design & development, Database management systems (DBMS), Teaching Case

1. INTRODUCTION

In the context of a traditional Structured Query Language (SQL) database management course, students will learn concepts and skills geared around typical SQL relational databases, such as entities, relationships, normalization, and data structuring. They also learn SQL is typically used to create tables and store data into databases, and that they will need to query data in order to build knowledge from them. Once the students learn how to query traditional databases, they will be introduced to a NoSQL database management system. NoSQL is a relatively new technology that many companies use to collect big data, which is quickly becoming an integral part of modern information systems. The low cost of data storage coupled with cloud based infrastructures allows not only the major, multinational organizations, but also small and medium enterprises to utilize big data (Purcell, 2014). Big data is also becoming an essential aspect of many of today's web applications. Social media sites are storing massive amounts of data on a daily basis. For instance, Facebook logs 135 billion messages per month, and Twitter stores seven terabytes of data every day.

Some of the challenges in the use of big data are in identifying how to handle the velocity at which data are collected and deciding the most effective way to store the data. In many cases, traditional SQL databases cannot process data fast enough. In other cases, traditional relational database structures are not flexible enough to handle the varying data types and the inconsistent manner in which the data are collected. To deal with these challenges, major Internet companies, like Google, Facebook, Twitter, and Amazon, are moving to NoSQL databases. The term, NoSQL, coined in the late 90's to mean "Not Only SQL," describes non-relational database systems (Berg, Seymour, and Goel, 2013). NoSQL databases allow for large amounts of variable and unstructured data to be stored and used. Common uses for these databases are session stores, user profile stores, mobile apps, e-commerce, and third-party data aggregation (Meijer and Bierman, 2011; Moniruzzaman and Hossain, 2013; Stonebraker, 2010).

Because of the challenges of big data, Information Systems (IS) students need to develop a basic understanding of how NoSQL databases work in modern information environments. This need is especially significant with the adoption of cloud-based infrastructures, which makes using this kind of database very likely during their careers. So, a clear understanding of what a NoSQL database looks like, how it differs from a traditional SQL database, and how to use the non-traditional database is necessary for IS college majors in the near future. This paper presents a teaching case for adding a NoSQL lesson to an existing traditional relational database course.

This NoSQL teaching case includes instructions on integrating the teaching module into a traditional database management course. Incorporating this NoSQL teaching case into both undergraduate and graduate database management courses will give students experience with this emerging technology. The case uses CouchDB (Couchdb.apache.org), an open source NoSQL database management system software that students can either install on their personal computers or use as a cloud-based service.

2. A BRIEF REVIEW OF THE LITERATURE

Goes (2014), editor-in-chief of *MIS Quarterly*, identified an industry gap in the understanding of big data, its challenges, and its potentials. In the past several years, big data has emerged as a new trend in the field of information technology because of the increasing amounts of data organizations in all sectors are generating, collecting, storing, and analyzing. Goes further explains IS groups are acknowledging this industry gap by offering academic programs specializing in data and business analytics that develop students' skills in data science.

The term, big data, describes datasets so large and complex that they are difficult to process using traditional processing applications. Plus, these large and complex databases consist of both structured and unstructured data and may come from traditional or new data sources. Another important aspect is that big data size is not a set value, but is an ever-increasing, moving target. Over the years, the scope of big data has expanded significantly. Now, big data not only refers to the data itself, but also to a set of data management technologies that social media companies first used to capture, store, manage, and analyze massive volumes of data (Shaw, 2014).

Stainer (2012) explored introducing the basic concepts of NoSQL in a traditional database curriculum by discussing how they differ from relational databases and what issues there may be in introducing NoSQL into a traditional database course. She found that instructors are already challenged in covering the breadth of information pertinent to modern databases and that the prevalence of using databases for business analytics increases this challenge. However, Stainer also recommended that database course instructors incorporate NoSQL into the curriculum teaching as a way to follow the pace of the database industry. To introduce NoSQL concepts to the database curriculum, Stanier (2012) proposed a teaching strategy in which she takes a relational scenario, like a student/module example, and executes the scenario in both a relational and a NoSQL document datastore. This way, her students are able to compare the different way data are handled and identify issues with the design. Students then spend time generating and assessing a NoSQL datastore for an incidents log in which each document has a different structure, where the structure of the log is not limited by the requirements of a logical schema and where document structure may be changed on the fly. Stanier explained that introducing NoSQL in the database curriculum is a necessary response to industry changes, however doing so brings with it many challenges in database courses.

Because traditional SQL databases have difficulties processing large, unstructured datasets, NoSQL databases have become an alternative. Thus, teaching cases have become valuable resources for information system educators (Cappel and Schwager, 2002; Hackney, McMaster, and Harris, 2003). Over the past few years, there have been a number of teaching cases published on various IS emerging technologies (Boyle, 2007; Olsen and Hauser, 2007; Winkelmann and Leyh, 2010; Xu, Rondeau and Mahenthiran, 2011). However, the review of literature (Padhy, Patra, and Satapathy, 2011; Paterson and Gray, 2014) reveals there are few published NoSQL teaching cases that can be used to introduce NoSQL in an undergraduate course. This gap in the literature justifies the need for a teaching case that will help students understand how the NoSQL technology is used to manage the more unstructured big data demands.

3. THE CASE

Students will create a NoSQL database for a social media webpage assignment. They will use CouchDB as their database management system to create a database from a textual case for populating social media pages.

3.1 Scenario

The administration at your school, along with the Student Government Association (SGA), feel that a stronger student community will create happier students, which will lead to better student retention and, ultimately, better graduation percentages. So, they have decided to create a private social media website for the campus. They believe this site will help students connect with other students and learn more about each other, which will ultimately bolster attendance for school functions, create opportunities for peer collaboration, and support campus diversity. They have asked you to create a database to support the site. Since you know you want to collect all the data that each student feels represents him or herself, you have decided you need to use a NoSQL database. Thus, you will need to create a new database and define the fields and values, so you can collect relevant data from each student.

3.2 Installing CouchDB – NoSQL DBMS

Students can either install CouchDB on their personal computers or use it as a cloud-based service. For this lesson, students will use CouchDB as their Database Management System. CouchDB is an open-source DBMS provided by the Apache Software Foundation. It uses JavaScript Object Notation (JSON) to store data and JavaScript as its language for queries. CouchDB is a document-oriented, or document-store, database. In this style of NoSQL database, all data are stored in a "document" and identified by a unique key. Fields within a NoSQL document can vary from key to key, as opposed to a relational database in which fields must be consistent among records or keys. Other suitable NoSQL database systems for use with the data set are MongoDB or Couchbase.

3.3 NoSQL Data Set

Instructors will give students a textual case of information that represents possible data for users wanting to populate their social media page. The students will need to read the case to identify the necessary data to be entered into the database. The textual case will also determine what fields and values students must create in their database.

In the textual case, the students will receive information on several users. That information will include data consistent with social media data, such as name, address, email, gender, interests, and friends. Some data will be consistent from person to person. Other data will vary from no data to many data points. Instructors can use the example relational database set to illustrate what this type of data set would look like in a relational table to illustrate its flaws.

3.4 Social Media Page

The students will create a database through CouchDB based on the information obtained from the textual case as well. Once they create the database, they will enter each person's data into a new document, and each document will store the data with a field and value. An example of a field and value would be "name" as the field and "John Doe" as the value, or "interests" as the field and "running, games, bikes" as the value. The instructor can also elect to allow the students to create other fields that they think will be useful for a social media page.

3.5 Queries

CouchDB uses JavaScript as its querying language. Compared to traditional SQL, NoSQL querying can be very complex; however, a few simple queries will be provided for the student to get a basic understanding of how to get data out of a NoSQL database.

Queries are executed in the Views section of the CouchDB interface. To make a query in CouchDB, select "Temporary View" from the "View" dropdown window. There will be a default function in the "Map Function" window, which is shown in Figure 1. With this function, the students will write the code to return a null response for the "Key" and all data in all documents for the database the students created for their school's social media page.

```
function(doc){
      emit(null, doc);
}
```

Figure 1. NoSQL Function to Query the Database to Show all Data

Figure 2 provides the code students are asked to write to return the "name" fields for the Key and the corresponding "address" fields for the "Value" for all documents in the social media page database.

```
function(doc){
      emit(doc.name,doc.address);
}
```

Figure 2. NoSQL Function to Query all Names and Addresses in the Database

Next, students will write code to return the "name" fields for the Key and "address" fields for the Value for all documents that have male for their gender field, as shown in Figure 3.

```
function(doc) {
        if(doc.gender == "male") {
        emit(doc.name,
 doc.address);
        }
}
```

Figure 3. NoSQL Function to Query all Names and Addresses if Gender is Male

3.6 Implementing the Case

Instructors will start the NoSQL module of the database course by giving the students a pre-test to assess their prior NoSQL knowledge. They will then teach NoSQL's background and its importance. In this section of the module, the instructors will give the students a scenario that required a NoSQL database as well as sample data and guide them in implementing the NoSQL queries. Then, the students will go through the CouchDB "getting started" tutorial and learn how to use the software. The instructors will then give each student an independent assignment asking her/him to create new NoSQL documents. Following their independent work, the instructors will administer a post-test to assess the knowledge they have gained in the lesson as well as their perceptions of the teaching case.

4. DATA ANALYSIS AND RESULTS OF CASE IMPLEMENTATION

This teaching case has been tested in two undergraduate database management courses.

4.1 Demographics of the Participants

The teaching case was administered to 44 undergraduate students in a traditional database management course at a state liberal arts university. The demographics of the students are provided in Table 1 below. The average age of those surveyed was 20.86 years (SD = 1.26). The majority of the participants reported that they had moderate to very strong computer experience, (M = 5.5, SD = 1.38).

Characteristics		Frequency
Gender		
	Male	31
	Female	12
	Did not report	01
Age		
	19	03
	20	13
	21	23
	22	02
	24	02
	26	01
Major		
	Accounting	03
	Management	05
	Marketing	06
	Business	01
	MIS	27
	Did not report	02

Table 1: Demographics of the Participants

4.2 Pre- and Post-Test Results

The students' basic knowledge of NoSQL, as shown in the pre-test, was quite low, (M=40.78), (SD=34.90). However, the post-test revealed their general NoSQL knowledge increased greatly, (M = 87.92), (SD = 16.15). Following the teaching case, the students' understanding of NoSQL popularity among developers of social networking website increased, (M = 0.53). Likewise, the students' knowledge that NoSQL databases provide an environment that easily accepts unstructured data increased, (M = 0.57).

5. CONCLUSION

There are many reasons for choosing to implement NoSQL instead of relational databases, and these reasons can be as various as the volume of data collected in a given time frame to the unique quality of the data. In our teaching case, we wanted to use a NoSQL database for the purpose of storing semi-structured data, and one of the best examples of semi-unstructured data are social media pages. Since many students use social media, the choice of this scenario works well in the classroom. Another reason we chose social media pages was because the collection of data on the users' preferences, as opposed to the developer's preferences. This significant shift means the amount and type of data entered into a social media page depend completely on the user, and the user may choose to share a little information or a lot. Additionally, the database needed to be able to store this semi-structured data in a way that allows the social media page to use the data to enhance the page's value. This teaching case, then, has the two-fold benefit of teaching students how social media handles their data while preparing them for using NoSQL databases in the future.

The NoSQL teaching case works well since most students use social media tools. It also offers instructors the opportunity to integrate a lesson on collecting big data at the same time the students are exposed to timely concepts about database applications in the modern technology environment. They learn why NoSQL is a solution for processing a large dataset, how their data are handled in their social media pages, and how to begin to use NoSQL databases in the future. Developing students' understanding of big data's reach, as well as the almost limitless potential for its analysis and application, is a growing necessity for educators concerned with students' information literacy and future earning potential.

6. REFERENCES

Berg, K. L., Seymour, T., & Goel, R. (2013). History of Databases. *International Journal of Management & Information Systems*, 17(1), 29-36.

Boyle, T. A. (2007). Computers-for-edu: An Advanced Business Application Programming (ABAP) Teaching Case. *Journal of Information Systems Education*, *18*(3), 283-296.

Cappel, J. J. & Schwager, P. H. (2002). Writing IS Teaching Cases: Guidelines for JISE Submission. *Journal of Information Systems Education*, 13(4), 287-294.

CouchDB [Software]. Available from Apache CouchDB: http://couchdb.apache.org

Goes, P. B. (2014). Big Data and IS Research. *MIS Quarterly*, 38(3), iii-viii.

Hackney, R. A., McMaster, T., & Harris, A. (2003). Using Cases as a Teaching Tool in IS Education. *Journal of Information Systems Education*, 14(3), 229-234.

Meijer, E. & Bierman, G. (2011). A Co-Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 54(4), 49-58.

Moniruzzaman, A. M. & Hossain, S. A. (2013). NoSQL Database: New Era of Databases for Big Data Analytics – Classification, Characteristics and Comparison. *International Journal of Database Theory and Application*, 6(4), 1-14.

Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011). RDBMS to NoSQL: Reviewing some Next-Generation Non-Relational Databases. *International Journal of Advanced Engineering Science and Technologies*, 11(1), 15-30.

Paterson, J. H. & Gray, E. (2014). Teaching NoSQL with RavenDB and Neo4j. *12th International Workshop on the Teaching, Learning and Assessment of Databases*, Southampton Solent University.

Purcell, B. (2014). Big Data using Cloud Computing. *Journal of Technology Research*, 5, 1–8.

Shaw, J. (2014). Why "Big Data" is a Big Deal. *Harvard Magazine*, 3, 30-35.

Stanier, C. (2012). Introducing NoSQL into the Database Curriculum. *10th International Workshop on the Teaching, Learning and Assessment of Databases*, The Open University.

Stonebraker, M. (2010). SQL Databases v. NoSQL Databases. *Communications of the ACM*, 53(4), 10-11.

Winkelmann, A. & Leyh, C. (2010). Teaching ERP Systems: A Multi-Perspective View on the ERP System Market. *Journal of Information Systems Education*, 21(2), 233-240.

Olsen, D. & Hauser, K. (2007). Teaching Advanced SQL Skills: Text Bulk Loading. *Journal of Information Systems Education*, 18(4), 399-402.

Xu, H., Rondeau, P. J., & Mahenthiran, S. (2011). The Challenge of Implementing an ERP System in a Small and Medium Enterprise: A Teaching Case of ERP Project Management. *Journal of Information Systems Education*, 22(4), 291-296.

AUTHOR BIOGRAPHIES

Brad Fowler is a Lecturer in the Computer Science & Information Systems department of the J. Whitney Bunting College of Business at Georgia College and State University. He received his B.A. degree in Music Performance based in Jazz Studies from the University of Alabama and holds a Master's degree in Management Information Systems from Georgia College & State University. Fowler teaches Principles of Information Systems, Intro to IT, Desktop Publishing, and Web Content Tools. His research interests include enterprise resource planning systems, micro-organizations, NoSQL, SaaS platform support. He is also interested in learning, developing, and applying dynamic teaching methodologies that engage the students in and outside of the classroom.

Joy Godin is Assistant Professor of Management Information Systems in the J. Whitney Bunting College of Business at Georgia College & State University. She teaches both undergraduate and graduate students various courses including Database Management, Introduction to Enterprise Resource Planning Systems, Programming, and Principles of Information Systems. Godin's research interests include virtual teams, collaboration technologies, technology acceptance, online learning, project-based learning, database technologies, and analytics.

Margaret Geddy is a Business Communications lecturer in the J. Whitney Bunting College of Business at Georgia College & State University. In addition to her years of academic instruction at the university level, she has also worked in the private sector as a Business Content Editor. Geddy's research interests include interdisciplinary writing across the business curriculum, project-based writing, and student engagement in writing beyond the classroom.

aitp Association of Information Technology Professionals

No matter how sophisticated the technology, it still takes people! ™

EDSIG

Serving Information Systems Educators

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 1055-3096