3.2.2 Complementing CS DB with IT.

It is common that students in IT database courses are also taught specific databases. [13] shows that teaching a specific database, such as Oracle, and SQL together can help students better understand ANSI-standard SQL. By learning how an implementation can be deviated from SQL standard, students would not only better understand the general SQL knowledge, but be better prepared to use any specific database after the class.

We would suggest that the CS database course, while still focusing on teaching general database concepts and techniques, could benefit from including topics related to specific database systems. In addition to learning a concrete implementation of SQL standard, students can benefit from learning specific data management features offered by individual databases. For example, Oracle supports optimization tuning, allowing users to proactively monitor a system's performance, analyze statistics related to database, application, operating system, network, and disk I/O, and tune performance through the SQL tuning advisor. Introducing those topics into CS courses would help students better understand data queries through the link from those queries to their concrete executions and help them learn the strategies of improving the quality of those queries.

4. DISCUSSION

From the above analysis, we have reached several important observations. We first summarize these observations and then offer our suggestions on how the IT and CS curricula can benefit from each other, which may help shape the future IT and CS curricular development.

First, both CS and IT students start to learn a similar set of foundational topics in the entry level courses and then deviate from each other when it comes to more advanced topics. For example, the topic list of the introduction to database courses from both CS and IT shows a significant overlap with over 90% of topics appear in both lists. This is reasonable as students need to get to know the basic vocabularies, fundamental concepts, and standard functionalities to gain a good overall understanding on how a database management system (DBMS) works. However, the focus becomes very different in an advanced elective course in database. In the database implementation course that is usually offered to CS students in their senior year, topics show a strong algorithmic flavor. In many institutions, CS students need to implement a small-scale DBMS or its major technical components. In contrast, the advanced elective database course in the IT curriculum (e.g., the Database Management and Access course as discussed in this paper) typically focuses on the practical usage of a popular DBMS, such as Oracle or SQL Server. In such a course, students are usually exposed to a popular DBMS that is commonly used in the industry and gain practical skills of how to use and manage the system. The distinct training that students receive from these two different curricula may help determine the workforce readiness and lead to different career paths for IT and CS students, which will be discussed later in this section.

Second, the delivery methods also tend to be different. In particular, hands-on exercises usually play an important role when teaching courses in an IT curriculum. The difference in the teaching methodology is essentially driven by the different focus of the IT and CS curricula. As discussed in Section 2, IT courses tend to have a more practical focus as compared to CS courses which are more theoretical. The difference in the focus can be directly reflected from the actual subjects covered in the corresponding courses. For example, in the advanced database courses as discussed above, IT courses usually cover specific technologies through one or two popular DBMSs while CS courses focus more on the fundamental building blocks and their foundational underpinnings and tend not to discuss any specific database products. Additional evidence can also be found in the published SIGITE and SIGCSE papers. Most of the database related SIGITE papers discuss specific technologies or database products, including Oracle, SQL Server, and .NET while corresponding SIGCSE papers tend to discuss general concepts/theories, such as XML, query simulation/execution, and design patterns, which are not restricted to any specific system or product. A key benefit of the hands-on delivery of teaching materials is that students learn the best practice from the course instructors and get exposed to the right way of doing things. By following the course instructor's well-designed examples, students can effectively avoid making many common mistakes. However, one potential downside of such teaching approach is that students may get used to following course instructors' way of doing things instead of being challenged to develop their own version of best practice through a trial-and-error process. In addition, as the hands-on practice will take a large portion of the class, students may not have the opportunity to learn more and deeper concepts that can further extend their knowledge.

Third, the difference in the curricular content and teaching methodology will lead to IT and CS graduates with quite different skill sets. In particular, for the IT graduates, due to the practical focus of the IT curriculum and the rich hands-on experience that they have developed during the learning process, they are very well prepared for the job that they are hired to do and require little or no additional training. As a result, IT graduates have a high reputation in terms of their workforce readiness. In contrast, most CS graduates need some transition time and/or training when moving from school to their job environment. Since the CS curriculum primarily focuses on the underlying theories and the general foundation, it is usually hard for CS students to develop systematic knowledge on a specific system or product. Nonetheless, since the theories and foundation are typically shared across different systems and products, after the transition and training, most CS graduates will be able to perform very well in their jobs. Furthermore, thanks to their deeper knowledge in various subjects and systematic training and practice in algorithm development, CS students usually demonstrate stronger capability to adapt to new technological advances, which can significantly benefit their career path in a longer term.

Based on the above observations and our own experience as IT and CS educators, we would like to offer some suggestions that

can help improve the design of future IT and CS curricula and better prepare graduates from both programs for their feature career. First, it appears that complementing both the course content and the teaching methodology from IT and CS will bring additional benefit to both IT and CS students. In fact, in the authors' institution, there are always a good number of CS students in each term that choose to take some IT courses and the same situation also applies to the IT students. In essence, students are trying to using their own ways to receive complementing training from both IT and CS curricula. Second, extra caution should be given when choosing topics for complementing these two curricula because it is still of critical importance to keep the unique characteristics of each curriculum so that they can properly serve students with different background and interest. Ultimately, IT and CS graduates should not possess completely overlapping skills so that they can serve different needs from their future employers. Last but not least, the complementing design should provide enough flexibility that accommodates diverse needs from students with different background and skill sets. It may be beneficial to follow a modular design that groups similar topics into course modules, where some modules may have a theoretical focus and other may have a practical one. Given the needs and requirements from the students, different modules can be offered in a flexible way. Some courses could be intentionally designed to offer to both CS and IT students, where they have the opportunity to work together on some team project to learn from each other. Such team projects will also help better prepare students for their future working environment, where people with different skills collaborate to accomplish a large-scale project.

5. CONCLUSION

In this paper, we use database courses offered by both IT and CS curricula as a case study to identify commonalities and differences between these two important computing disciplines. Our analysis further helps reveal the root reason that causes those differences, which eventually show significant impact on the graduates' future career paths. In particular, a strong emphasis on the practical skills better prepares IT graduates for their workforce readiness while a deeper theoretical knowledge base puts CS graduates in a better position to adapt to technological advances that benefits them in a longer term. Given the identified differences in terms of both topics and teaching methodologies, we provide important recommendations that may help shape the future IT and CS curricular development.

REFERENCES

[1] Adnan Abid, Muhammad Shoaib Farooq, Ishaq Raza, Uzma Farooq, and Kamran Abid. 2015. Variants of Teaching First Course in Database Systems, Bulletin of Education and Research, December 2015, Vol. 37, No. 2 pp. 1-17.

[2] ACM/IEEE-Curriculum 2001 Task Force. Computing Curricula 2001, Computer Science. IEEE Computer Society Press and ACM Press, December 2001. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2001.pdf.

[3] ACM/IEEE-Curriculum 2005 Task Force. Computing Curricula 2005, Computer Science. IEEE Computer Society Press and ACM Press, September 2005. https://ieeecs-media.computer.org/assets/pdf/CC2005-March06Final.pdf.

[4] Brett Allenstein, Andrew Yost, Paul Wagner, and Joline Morrison. 2008. A query simulation system to illustrate database query execution. In Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08). ACM, New York, NY, USA, 493-497. DOI: https://doi.org/10.1145/1352135.1352301.

[5] Canada's Association of Information Technology (IT) Professionals, "A Guide to the Common Body of Knowledge for Computing and IT (CBOK)", Canadian Information Processing Society, Canada retrieved May 24, 2019 from http://www.cips.ca/CBOK.

[6] Computer Science Curricula 2013 - Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM), Dec. 2013.

[7] Costel Gabriel Corlăţan, Marius Mihai Lazăr, Valentina Luca, and Octavian Teodor Petricică. 2014. Query Optimization Techniques in Microsoft SQL Server. Database Systems Journal vol. V, no. 2/2014.

[8] Jean Habimana. 2015. Query Optimization Techniques - Tips for Writing Efficient And Faster SQL Queries. International Journal of Scientific & Technology Research Volume 4, Issue 10, October 2015.

[9] IT2008 Task Group. 2008. Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology. http://pin.epsig.uniovi.es/informatica/GradoII-TI/IT2008%20Curriculum.pdf.

[10] IT2017 Task Group. 2017. Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology. DOI: http://it2017.acm.org.

[11] Qusay H. Mahmoud, Shaun Zanin, and Thanh Ngo. 2012. Integrating mobile storage into database systems courses. In Proceedings of the 13th annual conference on Information technology education (SIGITE '12). ACM, New York, NY, USA, 165-170. DOI: https://doi.org/10.1145/2380552.2380602.

[12] Thomas J. Marlowe, Cyril S. Ku, and James W. Benham. 2005. Design patterns for database pedagogy: a proposal. In Proceedings of the 36th SIGCSE technical symposium on Computer science education (SIGCSE '05). ACM, New York, NY, USA, 48-52. DOI: https://doi.org/10.1145/1047344.1047375.

[13] Gary B. Randolph. 2003. The forest and the trees: using oracle and SQL server together to teach ANSI-standard SQL. In Proceedings of the 4th conference on Information technology curriculum (CITC4 '03). ACM, New York, NY, USA, 234-236. DOI: https://doi.org/10.1145/947121.947174.

[14] Han Reichgelt, Barry Lunt, Tina Ashford, Andy Phelp, Erik Slazinski, and Cheryl Willis. 2004. A comparison of baccalaureate programs in Information Technology with baccalaureate programs in Computer Science and Information Systems, Journal of Information Technology Education, 3, 19-34, 2004.

[15] Russell Shackelford, Andrew McGettrick, Robert Sloan, Heikki Topi, Gordon Davies, Reza Kamali, James Cross, John Impagliazzo, Richard LeBlanc, and Barry Lunt. 2006. Computing Curricula 2005: The Overview Report. In Proceedings of the 37th SIGCSE technical symposium on Computer science education (SIGCSE '06). ACM, New York, NY, USA, 456-457. DOI: https://doi.org/10.1145/1121341.1121482.

[16] Paul J. Wagner and Thomas K. Moore. 2003. Integrating XML into a database systems course. SIGCSE Bull. 35, 1 (January 2003), 26-30. DOI: https://doi.org/10.1145/792548.611924.

[17] Yanhao Zhu, James Crouch, and Mohammad H. N. Tabrizi. 2005. In-process object-oriented database design for .NET. In Proceedings of the 6th conference on Information technology education (SIGITE '05). ACM, New York, NY, USA, 139-142. DOI=http://dx.doi.org/10.1145/1095714.1095747.