Comparative Study of Relational Databases for CS Courses

Axel Luca
Computer Science
Worcester Polytechnic Institute
Worcester MA, United States
aluca@wpi.com

Ethan Pollack
Computer Science
Worcester Polytechnic Institute
Worcester MA, United States
epollack@wpi.com

Harrison Taylor
Computer Science
Worcester Polytechnic Institute
Worcester MA, United States
hstaylor@wpi.com

ABSTRACT

This paper assesses three leading database systems: Oracle, Postgres, and MySQL. The aim of this comparative study is to create systematic selection criteria for college instructors to assist them in choosing the most suitable database management system for teaching database courses. As many industries rely on databases, the choice of appropriate database systems for the curriculum is crucial to prepare students for real-world applications. Our methodology includes a literature review of each system, practical evaluations via installations on multiple operating systems, and syntactical analyses through hands-on projects. The results show unique features of each system, with implications for how they fit within university curricula, depending on course design and goals.

CCS CONCEPTS

• Information Systems → Data Management Systems; • Applied Computing → Education.

KEYWORDS

Database, Education, Structured Query Language (SQL), Complexity, Comparison

ACM Reference format:

Axel Luca, Ethan Pollack, and Harrison Taylor. 2023. Database Technologies Exploration and Comparison. In *Proceedings of ACM Student Research Competition for Special Interest Group on Computer Science Education Technical Symposium 2024 (ACM SIGCSE TS 2024). ACM, New York, NY, USA, 2 pages.*

1 PROBLEM AND MOTIVATION

With the proliferation of the internet and the technological field in general, the study of and requirements for data storage and management have become crucial to a solid computer science education. While raw text files may suffice for simpler applications, larger and more complex applications need the use of databases for efficient and secure data storage and manipulation. Managing these databases requires a Database Management System (DBMS). Despite the variety in DBMS availability, relational databases have proven to be the most popular. However, within this category, there are still many options to choose from to meet the needs for a specific application. Each comes with its own benefits and downsides, including, but not limited to, its dialect of Structured Query Language (SQL).

The primary motivation behind this research is to assist instructors in selecting the most suitable DBMS for their introductory courses. With the significance of database education in computer science curricula, it is important for educators to understand the tradeoffs of each of the available DBMS tools. Through this study's evaluation of the strengths and weaknesses of each relational database, instructors will be provided with the essential details to make informed curricular decisions. Additionally, this research aims to present students with enriched resources, ensuring they learn how to apply database fundamentals to real-world applications.

2 BACKGROUND AND RELATED WORK

Databases are a fundamental component of computer science education, with courses spanning introductory to advanced topics such as data modeling, query optimization, ER diagrams, views, triggers, procedures, functions, and relational algebra using systems like MySQL, Oracle, and Postgres [1, 2, 3, 4, 5, 6]. Their value is highlighted by the number and variety of industries that use databases, making it vital for computer science students to learn them well for jobs like software development and data analysis [3, 6, 7]. To ensure students' thorough understanding of database concepts, universities employ varied teaching methods based on their objectives. While some methods, like Ohio State University's interactive tools, prioritize conceptual clarity, others like Purdue and Worcester Polytechnique Institute emphasize real-world application through projects [2, 5, 7, 8, 9, 10]. Relational databases have been prominent in both industry and academia for decades, despite the existence of numerous other database subsets [11]. These databases, organizing data in tables with unique rows and columns, allow for enhanced productivity and ease of interaction, especially with Structured Query Language (SQL), which offers structured, secure, and effective data operations [12, 13].

3 METHODOLOGY

To evaluate the three relational databases MySQL, Oracle, and Postgres in the context of introductory database courses, we used a variety of criteria, typically gauging the usability and simplicity of the three languages. These criteria include but are not limited to

Installation and Setup Complexity, Proprietary Integrated Development Environments, Datatypes, Table Management, Data Manipulation, Query Simplicity, Advanced Functionality, Java Database Connectivity (JDBC), and Python Connectivity.

We evaluated the databases based on these criteria because they are all relevant to the learning process of an introductory relational database course. Ideally, the technological aspects should be as easy as possible for the students in the course, so they may focus on learning the relational schema logic and principles, rather than being bogged down with overwhelming syntactical intricacies [14].

The databases were evaluated using sample problem sets from a real introductory database course to most accurately represent the problems that students will encounter during these courses. We experimented with many different data sets, and tested functionality such as advanced table constraints, triggers, views, functions, and procedures.

4 RESULTS AND CONCLUSION

Category	**Oracle**	**Postgres**	**MySQL**
Installation and Setup	Difficult	Moderate	Moderate
Proprietary IDEs	Simple	Moderate	Simple
Datatypes	Simple	Complex	Complex
Table Management	Easy	Easy	Easy
Data Manipulation	Easy	Easy	Easy
Query Simplicity	Easy	Easy	Easy
Advanced Functionality	Easy	Moderate	Moderate
Java/Python Connectivity	Easy	Easy	Easy

Figure 1: Summary of comparisons across Oracle, Postgres, and MySQL

Oracle provides the simplest and most robust SQL dialect of the three technologies. The difficult SQL syntax is kept at a minimum, the datatypes are very simple, and the table and data management methods are intuitive and easy. The major downside of Oracle is the lack of compatibility with macOS, a popular operating system, which means that for anyone who uses a Mac computer, the setup is both more complex and more time-consuming.

Postgres was an outlier in many senses when compared to MySQL and Oracle. It has a wide variety of features that are applicable in many different circumstances and is incredibly versatile with its structure and functionality. However, a negative impact of this could be that it seems daunting to new students attempting to learn relational database concepts. The IDEs, Datatypes, and Advanced Functionality demonstrate this issue well, as though they are very versatile, they are more complex than the other two technologies being studied and may be daunting for newer students.

MySQL lands in the middle of Oracle and Postgres. The IDE is much simpler, like Oracle. MySQL is more like Postgres with respect to datatypes and installation processes. The datatypes can be somewhat complex; however, the installation is easy and supports many popular operating systems. The Advanced Functionality is more moderate, containing some similarities to Postgres and some to Oracle.

Overall, our findings do not suggest one language to be blatantly superior to the others or one to be particularly unusable, as each language has its own benefits and nuances, and none truly shone or was explicitly worse than the others.

ACKNOWLEDGMENTS

We would like to thank our advisors Professor Wilson Wong and Professor Rodica Neamtu for providing this project opportunity. Additionally, we extend our appreciation to the WPI Computer Science Department for teaching us the topics necessary to complete this project. Lastly, we would like to acknowledge our parents' unwavering support and encouragement.

REFERENCES

[1] Aggarwal, D., Winstead, C., & Tufte, K. (February, 2020). Leveraging Industry Benchmarks to Teach Database Concepts. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 1410-1410). https://doi.org/10.1145/3328778.3372562

[2] Udoh, E. (2006). Teaching database in an integrated oracle environment. In Working group reports on ITiCSE on Innovation and technology in computer science education (pp. 71-74). https://dl-acm-org.ezpv7-web-p-u01.wpi.edu/doi/pdf/10.1145/1189215.1189174

[3] Fekete, A. D., & Röhm, U. (2022). Teaching about Data and Databases: Why, What, How?. ACM SIGMOD Record, 51(2), 52-60. https://dl-acm-org.ezpv7-web-p-u01.wpi.edu/doi/pdf/10.1145/3552490.3552504

[4] Fekete, A. (2005, June). Teaching transaction management with SQL examples. In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (pp. 163-167). https://dl-acm-org.ezpv7-web-p-u01.wpi.edu/doi/pdf/10.1145/1384271.1384382

[5] Jiang, L., & Nandi, A. (April 2015). Designing interactive query interfaces to teach database systems in the classroom. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (pp. 1479-1482). https://dl.acm.org/doi/pdf/10.1145/2702613.2732900

[6] Bi, Y., & Beidler, J. (2008). Teaching database systems with web applications team projects. Journal of Computing Sciences in Colleges, 23(3), 82-88. https://dl-acm-org.ezpv7-web-p-u01.wpi.edu/doi/pdf/10.5555/1295109.1295130

[7] Deng, Y., Frankl, P., & Wang, J. (2004). Testing web database applications. ACM SIGSOFT Software Engineering Notes, 29(5), 1-10. https://dl-acm-org.ezpv7-web-p-u01.wpi.edu/doi/10.1145/1022494.1022528

[8] Rilett, D., & Russo, J. P. (2013). Using Amazon web services to teach web-enabled database concepts. Journal of Computing Sciences in Colleges, 28(6), 134-135. https://dl-acm-org.ezpv7-web-p-u01.wpi.edu/doi/pdf/10.5555/2460156.2460181

[9] Center for Project Based Learning» Project-Based Learning at WPI. (n.d.). Wp.wpi.edu. Retrieved September 28, 2023, from https://wp.wpi.edu/projectbasedlearning/proven-pedagogy/project-based-learning-at-wpi/#:~:text=Since%201970%2C%20project%2Dbased%20learning

[10] Project-Based Learning at WPI | PBL in Higher Education. (n.d.). Www.wpi.edu. https://www.wpi.edu/project-based-learning

[11] Dolezel, D., & McLeod, A. (2020). Big-Data Skills: Bridging the Data Science Theory-Practice Gap in Healthcare. Perspectives in health information management, 18(Winter), 1j.

[12] E. F. Codd. 1982. Relational database: a practical foundation for productivity. Commun. ACM 25, 2 (Feb 1982), 109–117. https://doi.org/10.1145/358396.358400

[13] Oracle. (2023). What is a relational database? Oracle.com. https://www.oracle.com/database/what-is-a-relational-database/

[14] DB-Engines. (2019). DB-Engines Ranking - popularity ranking of database management systems. https://db-engines.com/en/ranking