**ABSTRACT**

This paper describes ClassDB, a free and open source system to enable large-scale learning of data management. ClassDB is different from existing solutions in that the same system supports a wide range of data-management topics from introductory SQL to advanced “native analytics” where code in SQL and non-SQL languages (Python and R) run inside a database management system. Each student/team maintains their own sandbox which instructors can read and provide feedback. Both students and instructors can review activity logs to analyze progress and determine future course of action. ClassDB is currently in its second pilot and is scheduled for a larger trial later this year. After the trials, ClassDB will be made available to about 4,000 students in the university system, which comprises four universities and 12 community colleges. ClassDB is built in collaboration with students employing modern DevOps processes. Its source code and documentation are available in a public GitHub repository. ClassDB is work in progress.

**Author Keywords**

Data management; Scalability; Learning; FOSS; ClassDB

**ACM Classification Keywords**

H.2.4. Database Management: Database Applications; H.2.4. Database Management: Systems; K.3.2. Computers and Education: Computer and Information Science Education

**INTRODUCTION**

Data management has been a popular educational topic for a while, but the interest in it has exploded over the last few years following the rise of “data science” and “data analytics”. Some online environments [11, 12] already exist to help with learning data-management and are evidently scalable, but they tend to be limited to learning query languages and their workflow is typically designed for “self-learning” with no instructor involvement. Some data-management MOOCs [6] do include instructors in their workflow, but they tend to be either topic-specific or course specific.

ClassDB is an open learning system enabling students to learn a range of data-management topics from introductory querying to advanced “native analytics” operations where students write and run both SQL and non-SQL code (for example, in Python and R) directly inside a database management system (DBMS). More importantly, students can use the same environment to practice any topic, and switch between topics so they can **incrementally learn multiple related topics.** (Students rarely follow a sequential “waterfall model” of learning.)

Regardless of topics, each student has their own sandbox and instructors can read student work and provide feedback. By default, students have full control of their sandbox and instructors have read permission to student sandboxes. (See Figure 1.)

(We do not expand on it in this paper, but the system also permits students to form teams. It also permits “teaching assistants” to perform certain operations.)

ClassDB maintains detailed logs of user activity which helps students and instructors analyze progress and determine future course of action. For example, ClassDB maintains distinct chronologies of database connections, operations that create or remove data objects, and queries over existing data objects. Using ClassDB’s pre-defined views, students and instructors can relate these three sequences to determine what if any issue might exist in completing a particular task.

ClassDB was piloted in late 2017 by a class of university students. It is now being used in a second pilot after revisions and is scheduled for a larger trial later this year. After these trials, in early 2019, ClassDB will be made available to the entire Connecticut State University System for likely use by about 4,000 students (roughly 5% of total enrollment) spread over four universities and 12 community colleges.

ClassDB is currently available for instructors and can be integrated into other learning environments using the ClassDB application programming interface (API) [4]. In due course, we aim to make ClassDB a service for anyone to use without their own instance, while also maintaining the
ability for instructors and MOOCs to have their own instances.

ClassDB is built using free and open-source software (FOSS) and is itself FOSS. Further, it is built in collaboration with students employing modern DevOps processes, giving students first-hand experience in building large-scale systems. (Implementing and maintaining ClassDB is itself a learning opportunity, but that is not the focus of this paper.)

ClassDB is distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License [2]. Its source code, tests, and documentation are all available in a public GitHub repository [5].

The rest of this paper is organized as follows: The next section provides a brief technical overview of ClassDB in its current state. We then outline the work in progress and the work planned for ClassDB and its ecosystem. We then present summary remarks.

Being work-in-progress, ClassDB features are in various stages of design, development, and deployment. Where a feature’s deployment is imminent, for simplicity of writing, we humbly describe that feature in this paper as if it is already deployed. We describe all other features strictly in our vision for the future. (The exact state of the system is always verifiable in the public GitHub repository.)

CURRENT STATE

This section provides a brief technical overview of the current state of ClassDB. The overview covers only those aspects necessary to help readers appreciate the capabilities and possibilities of ClassDB as a learning system. Figure 1 shows the high-level architecture. The online documentation [4] has the complete technical details.

Using ClassDB, students can practice data-management in relational, XML, and JSON models, and they can import and export data in a variety of formats including text, CSV, and HTML. Students can presently express queries, functions, and analytical operations in SQL, XPath, XSLT, C, C++, Python, and R.

ClassDB includes scripts to easily add/remove functionality such as activity logging, non-SQL programming, and other advanced features. It permits any number of databases within a server, and each database includes many objects (tables, functions, and views) that make the database immediately usable. It exposes much of its functionality as API entry points which can all be invoked using only native tools. For example, the well-known, simple, and free client `psql` suffices when using ClassDB with PostgreSQL (Postgres) [10]. That is users, especially students, do not need to learn any new tool to use ClassDB. In fact, students do not at all notice ClassDB as they use the DBMS.

The ClassDB API includes declarative functions to add/drop students, teams, instructors, and teaching assistants. It also includes functions to transfer objects among users, and views to review activity and progress. The API also allows instructors to create data objects for students to consult and experiment with, with no practical limit on the size or nature of the objects. For example, an instructor can create shared tables for analytical operations on big data (with the tables really containing “big data”).

![ClassDB Architecture Diagram](image)

*A single server can manage thousands of databases. Each user can maintain many sandboxes; the sandboxes may be distributed over many servers/databases.*

ClassDB uses role-based access control [9] to manage students, teams, instructors, and teaching assistants. It uses database schemas for sandboxes and makes each student the owner of their schemas, giving them the ability to use the schemas for any learning activity. This includes the ability to permit other users to review work and provide feedback.

ClassDB can support thousands of simultaneous users and thousands of databases in each server instance, with the actual numbers generally limited only by the operating environment and infrastructure resources such as CPU, main memory, and disk space. More users and databases beyond that can be supported by simply adding server instances as necessary. (We expect to use a single server and a single database when ClassDB is opened to the entire university system for use by about 4,000 students.)

We attribute the scalability of ClassDB to three key factors:

- Industrial-strength DBMSs which already have the ability to support thousands of databases and users in a single server instance.
- Elastic infrastructure (“cloud”) to host DBMS instances.
- Design choices which fully exploit DBMS features such as role-based access control and schemas, and the use of well-known techniques such as load balancing and data encapsulation.

ClassDB is presently implemented as a Postgres application, but its architecture permits implementation in any DBMS. (We plan to implement a version for Microsoft SQL Server.) It is deployable in several situations including platforms such as Amazon Web Services [1] and Microsoft Azure [8].
WORK IN-PROGRESS AND PLANS
This section outlines the work in progress as well as the current plan for ClassDB and invites input from experts and the broader community of learning scientists.

We see the current ClassDB system as having proven the technical architecture and establishing the scalability necessary to support learning at large scale. We also see the current state as the “infrastructure layer” upon which an application ecosystem that forms a complete learning system can be (and needs to be) built. In a sense, the current state is akin to an operating system; the work in progress and future work is akin to building applications to run on the operating system.

The work in progress and the work planned can be placed into the following broad categories:

- Log analysis and feedback
- Exercise and tutorial integration
- Integration with other learning environments
- Infrastructure improvements

Log Analysis and Feedback
ClassDB already maintains several activity logs and provides many useful views of the logs, but we plan to develop better and deeper log analytics. We expect to develop some novel analytics, but for the most part, we expect to build on the work of other learning scientists.

The log analysis we plan are broadly of two kinds: those designed to help students assess and analyze their own progress; and those designed to help instructors assess student performance and progress. Ideally, we like to provide automated feedback and recommendations (Cummins and others [3] describe one such system), which requires deep learning over data collected from exercises and tutorials. (See the next sub-section.)

We also plan to expand the number and variety of logs, but a key concern with increasing logging activity is that too much of it can hinder system performance and potentially impact scalability. (Scalability is an obvious foundational need of a large-scale learning system such as ClassDB.)

Exercise and Tutorial Integration
ClassDB already permits instructors and students to add their own data. We plan to create an API to make it easier for instructors to integrate their own exercises and tutorials in to ClassDB as a first step toward a feedback system.

As part of this effort, we have created many exercises and tutorials for different topics and complexity levels and are using those exercises to understand the integration needs. We are also studying data-management MOOCs to understand additional scenarios.

Our experience shows that it is relatively easy to create a framework to integrate exercises and tutorials, but the real challenge is in creating an effective feedback system which in turn requires tight integration with logging.

Integration with other Learning Environments
Being an open learning system is one of our goals for ClassDB. This openness includes providing the ability for MOOC and other similar platforms to let their users practice data management using ClassDB.

Third parties may integrate their system with their own instance of ClassDB or integrate their system with an outside instance of ClassDB. Aside from authentication and other traditional system-integration concerns, both scenarios have comparable complexity.

ClassDB has been designed from the ground up for openness and its API already has many of the features necessary for third-party integration. However, there is work to be done to streamline the integration and authentication processes.

Infrastructure Improvements
As mentioned previously, the current state of ClassDB represents an “infrastructure layer” of sorts that supports the entire learning system. As is typical of any infrastructure layer, several improvements are possible, but we are interested in two key areas.

First, we plan to add support for additional DBMS brands: ClassDB is presently supported in Postgres and we intend to support it in Microsoft SQL Server (MSSQL). We choose MSSQL because it is popular and quite different from Postgres and other DBMSs, making it a worthwhile learning alternative.

The task of supporting ClassDB in multiple DBMS brands is non-trivial, but relatively straightforward. However, it can be challenging to mix multiple brands in a single deployment. Thus, our priority is to first add support for MSSQL, and later consider mixing multiple brands in the same deployment.

The second area of infrastructure improvements is to improve deployment in hosted environments, specifically in database as service such as those offered in Microsoft Azure [7]. The advantage of using such a service is cost reduction, but the challenge is the potential loss of server-level controls. (Database-as-a-service is designed to provide users a database without users concerning themselves with the DBMS.)

SUMMARY
To summarize, in its current state, ClassDB has adequately demonstrated that at an infrastructure level it is a large-scale open learning system for data management. It has been used in a pilot by university students, is presently in a second pilot, and will soon be made available to about 4,000 students in a university system comprised of 16 distinct institutions.

ClassDB supports learning a wide range of data-management topics from introductory querying to advanced analytics. Its workflow accommodates instructors, but it does not require
instructors. Lastly, it is an open system that can be integrated with other learning environments.

ClassDB is work-in-progress and we have plans for several “application level” additions. The additions are in three broad areas: log analysis and feedback, exercise and tutorial integration, and third-party integration. A fourth area of additions is at the infrastructure level.

ACKNOWLEDGMENTS
We are grateful to the students who participated in pilot uses of ClassDB. We also thank all contributors to ClassDB’s implementation, testing, and documentation. (A full list of contributors is available at the GitHub repository.)

REFERENCES
1. Amazon Web Services Inc. 2017. Amazon Web Services. Retrieved March 24, 2018 from https://aws.amazon.com/
2. Creative Commons. 2016. Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0). Retrieved March 24, 2018 from https://creativecommons.org/licenses/by-nc-sa/4.0/
3. Stephen Cummins, Alastair R. Beresford, Ian Davies, Andrew Rice. 2016. Supporting Scalable Data Sharing in Online Education. In Proceedings of the Third (2016) ACM Conference on Learning @ Scale, 97-100.
4. Data Science & Systems Lab. 2018. ClassDB Documentation. Retrieved March 24, 2018 from https://dassl.github.io/ClassDB
5. Data Science & Systems Lab. 2018. ClassDB Source Code. Retrieved March 24, 2018 from https://github.com/DASSL/ClassDB
6. edX. Data Analysis and Statistics Courses. 2017. Retrieved March 24, 2018 from https://www.edx.org/course?subject=Data%20Analysis%20%26%20Statistics
7. Microsoft Corporation. 2018. Azure Database for PostgreSQL. Retrieved March 24, 2018 from https://azure.microsoft.com/en-us/services/postgresql/
8. Microsoft Corporation. 2017. Microsoft Azure. Retrieved March 24, 2018 from https://azure.microsoft.com
9. PostgreSQL Global Development Group. 2017. Database Roles. Retrieved March 24, 2018 from https://www.postgresql.org/docs/10/static/user-manag.html
10. PostgreSQL Global Development Group. 2017. psql Documentation. Retrieved March 24, 2018 from https://www.postgresql.org/docs/current/static/app-psql.html
11. Teradata University Network. 2016. SQL Assistant. Retrieved March 24, 2018 from http://tunweb.teradata.ws
12. W3Schools. 2018. SQL Tutorial. Retrieved March 24, 2018 from https://www.w3schools.com/sql/