Database education is at an inflection point. With the surge of interest in all things "data", enrollments in traditional database courses are at an all time high. At the same time, the rise of Data Science as a discipline has led to the creation of new courses whose content significantly overlaps that of an introductory database course (e.g. data preparation, cleaning, SQL). Students from all across campus aspire to take data science courses, even with limited Computer Science backgrounds. This juxtaposition of content and proliferation of audiences is causing many database educators to question what we should be teaching in our data-oriented courses, and what resources we should use to teach them.

This guest column in SIGMOD Record will present a series of perspectives on database education, and in particular how it relates to a Data Science curriculum. The first episode is a summary of a panel on this topic at the 2021 VLDB Conference -- "The Future of Data(base) Education: Is the Cow Book Dead?" [1], which focused on three questions: What is the role of the database field in "computing"? What should we teach in a modern database course? And how do we place ourselves relative to Data Science?

Susan B. Davidson
University of Pennsylvania

VLDB Panel Summary: "The Future of Data(base) Education: Is the Cow Book Dead?"

Zack Ives
University of Pennsylvania
zives@cis.upenn.edu

Johannes Gehrke
Microsoft
johannes@microsoft.com

Jana Giceva
Technical University of Munich
jana.giceva@in.tum.de

Arun Kumar
University of California, San Diego
arunkk@eng.ucsd.edu

Rachel Pottinger
University of British Columbia
rap@cs.ubc.ca

1. What is the role of the database field in "computing" moving forward? Are we "the Data People" or "Yesterday's News" within the context of ML, big data, and data science?

The panelists agreed that databases remain central within computing, and that our ideas have had a lot of impact.

From an economic perspective, today's database market is roughly $60 billion and is expected to double in the next five years to about $120 billion (depending on who you ask). There is huge investment by cloud companies in data infrastructure, and even more to come. There's also a ton of innovation in the startup ecosystem, for example, Snowflake and DataBricks. And people with our skills are in such high demand in industry that we can barely graduate enough students.

The core ideas in databases have had a lot of influence. Relations, relational algebra, and aggregate queries are

found in Pandas Dataframes. Indexing and sharding are key ideas within data processing, and Map-Reduce is a central parallel data processing framework. Taking a program in a declarative, high-language and creating an efficient computation plan out of it is at the heart of everything in data science.

If today we appear to be the "poor cousins" of ML and AI (which have themselves gone through some ups and downs in popularity over the years), it is partially because we need to be more effective story tellers. For one of the first times in computing history, we have the opportunity even in our introductory computing courses to tell stories of data, and the beauty of data and computing. We can show how databases have added real economic value by changing the landscape of industry, and expose our students to interesting new challenges in modern data management. We can also talk about the rich future of applications and ideas from our field, for example, exciting new research ideas to create novel tools and systems that are relevant for ML and data science practitioners.

2. What should we teach in a modern database course for a computer science student? What aspects of this course should be integrated into a data science curriculum?

There was general agreement that there are more ideas than can fit in a single database course, but there was less agreement about what the actual topics should be in the first database course for computer science majors. The relational model, algebra, and SQL are foundational, although pieces of this have moved into other courses (in particular, introductory programming and data science courses). Relational design is important, although relational design theory is tough for some audiences and possibly not relevant for others. Indexing and performance are also fundamental, although modern computers look very different today from when our introductory database and database systems textbooks (such as the "cow" book [2]) were written. Transactions, concurrency control and recovery were considered by some to be either "irrelevant" or out-of-scope for an introductory course, but others considered it still important to teach at least what transactions are and how to use them. Data privacy and security are also increasingly important, but perhaps part of a more advanced course. Overall, the feeling was that many of the techniques and ideas of our introductory database textbooks are still quite relevant, and that some subset of these ideas should be included in an introductory course for computer science majors, but that they need to be rethought in the context of cloud computing and cloud data science infrastructure. Additionally, it may make sense to spend more time talking about the database management system as one component in a larger data management and processing ecosystem, rather than as the entity responsible for managing and processing all data.

Since there are more ideas than can fit in a single course, several panelists and audience members described their programs, which include a sequence of courses on databases, such as: introduction to databases, database applications, and database systems implementation. However, the sense was that the upper level course on database systems implementation should be rethought of more generally as "data infrastructure systems". Some felt that this course should also add three new topics: cloud computing, machine learning for database management systems, and database management for machine learning.

Other programs tailor the sequence depending on the audience, creating a subtree of courses. As an example, at UCSD there are database courses within the computer science and engineering department (CSE) as well as within the newly formed Halicioglu Data Science Institute (HDSI). CSE's "Database System Principles" covers relational model+algebra, SQL, normalization theory, and transaction theory, and is the gateway to other courses: applications, database systems implementation course, and an online analytics course. HDSI's "Intro to Data Management" drops normalization theory and transactions and instead emphasizes DataFrames, Pandas, and SQL, including some physical optimization capabilities. The follow-on course, "Systems for Scalable Analytics" discusses parallelism, scalability, cloud computing, Spark, dataflow systems and ML systems, and what's behind optimization.

However, the ability to offer more than one course depends on the available faculty resources at an institution. Whether a database course is part of the core computer science curriculum at an institution also varies, with most programs in Europe requiring it and most programs in North America letting students "vote with their feet".

3. If computer science and data science are separating as fields, where do we place ourselves?

Many panelists (and participants) felt that the database field is a "bridge" between computer science and data science, much as computer architecture is a bridge between electrical engineering and computer science. Machine learning and natural language processing are also bridges, but few other computer science topics are as fundamental to the data science process of data acquisition, processing, cleaning, visualizing, and analysis.

Another viewpoint is that we are fundamentally computer scientists. The database community provides the tools and skill sets that data scientists need to use data to answer their (domain-specific) questions. We develop the systems/concepts/abstractions that are used in data science. This includes fundamental ideas in many different areas, including declarative languages, normal forms, and transactions.

The growth of data science has also forced us to expand our focus beyond storing all the data in a single platform that does the management. In the past, database systems have been very siloed from other tools, and tend to own the whole processing stack. Thought has been put into integrating a Web server over a database, but apart from that a database is not really part of a broader data processing pipeline. As new data types are encountered, they are added to the data model. As new functionality is needed over that data, user defined functions are added to the language. In contrast, one of the reasons for the popularity of dataframes and Pandas is that they move between processing steps: they can be used to interact with databases, visualization packages, as well as with data analytics. We need to expand our focus beyond storing all the data in a platform that does the managing to a broader view of data infrastructure systems, in which the concepts and abstractions that we use (including indexing, optimization, transactions, provenance, etc) are applied.

Closing thoughts

We have been very well served by the database textbooks written by members of our community, e.g. [2-5], since they focus on fundamentals that apply across changes in hardware and operating environments. However, there are new ideas that should be added, and the existing ideas need to be updated in the context of cloud data infrastructure.

At the same time, communicating knowledge is moving beyond textbooks. Many of us, especially during COVID, have recorded lecture segments. YouTube videos are available on many of the topics covered in a database course. Sample homeworks and exercises are available on the web, whether we have approved it or not. Online tutorials are widely available for many of the newer technologies that we use to teach the fundamental ideas, and these technologies will change faster than we can write books. An interesting question is whether we as a community can more effectively share our teaching resources, and how we can use these resources to fill in knowledge gaps for our students.

From the high level of engagement from the panel and the audience, it was clear that the members of the database community are very invested in educating students on databases and data science, that they are eager to explore alternative answers to the questions above, and that they are interested in working together to tackle these problems.

Acknowledgments

We would like to thank members of the audience who contributed many good ideas through chat. These ideas have been incorporated into the summary.

References

[1] Zachary G. Ives. The future of data(base) education: Is the "cow book" dead?. PVLDB, 14(12): 2021. doi:10.14778/3436905.3436909. YouTube video: https://www.youtube.com/watch?v=DOWumD2UpOQ&ab_channel=VLDB2021.

[2] Raghu Ramakrishnan and Johannes Gehrke. Database Management Systems, 3rd Edition. McGraw Hill (2003)).

[3] Avi Silberschatz, Henry F. Korth, and S. Sudarshan. Database System Concepts, 7th Edition. McGraw Hill (2019).

[4] Ramez Elmasri and Shamkant Navathe. Fundamentals of Database Systems, 7th Edition. Pearson Prentice Hall (2015).

[5] Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom. Database Systems: The Complete Book, 2nd Edition. Pearson Prentice Hall (2008).