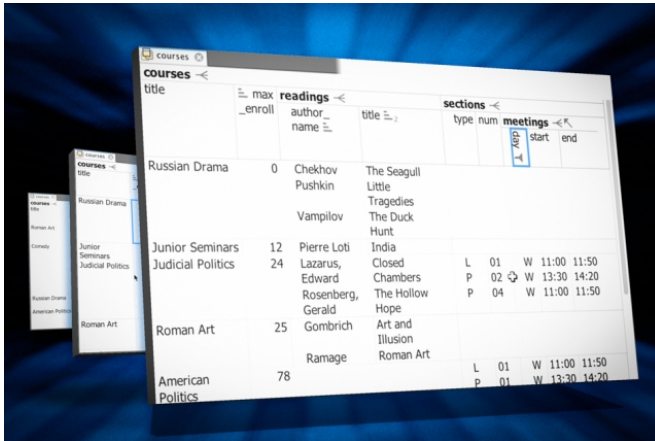


# New database interface looks like a spreadsheet

11 July 2016, by Larry Hardesty



Simple drop-down menus let the user pull data into the tool from multiple sources. The user can then sort and filter the data, recombine it using algebraic functions, and hide unneeded columns and rows, and the tool will automatically generate the corresponding database queries.

The researchers also conducted a usability study that suggests that even in its prototype form, their tool could be easier to use than existing commercial database systems that represent thousands, if not tens of thousands, of programmer-hours of work.

A new database interface looks like a spreadsheet. Users don't need to know database scripting languages such as SQL to create new database queries. Credit: Database images courtesy of the researchers (edited by MIT News)

When an organization needs a new database, it typically hires a contractor to build it or buys a heavily supported product customized to its industry sector.

Usually, the organization already owns all the data it wants to put in the database. But writing complex queries in SQL or some other database scripting language to pull data from many different sources; to filter, sort, combine, and otherwise manipulate it; and to display it in an easy-to-read format requires expertise that few organizations have in-house.

New software from researchers at MIT's Computer Science and Artificial Intelligence Laboratory could make databases much easier for laypeople to work with. The program's home screen looks like a spreadsheet, but it lets users build their own database queries and reports by combining functions familiar to any spreadsheet user.

"Organizations spend about \$35 billion a year on relational databases," says Eirik Bakke, an MIT graduate student in [electrical engineering](#) and computer science who led the development of the new tool. "They provide the software to store the data and to do efficient computation on the data, but they do not provide a user interface. So what inevitably ends up happening when you have something extremely industry-specific is, you have to hire a programmer who spends about a year of work to build a user interface for your particular domain."

## Familiar face

Bakke's tool, which he developed with the help of his thesis advisor, MIT Professor of Electrical Engineering David Karger, could allow organizations to get up and running with a new database without having to wait for a custom interface. Bakke and Karger presented the tool at the Association for Computing Machinery's International Conference on Management of Data last week.

The tool's main drop-down menu has 17 entries, most of which—such as "hide," "sort," "filter," and "delete"—will look familiar to spreadsheet users. In the conference paper, Bakke and Karger prove that those apparently simple functions are enough to

construct any database query possible in SQL-92, which is the core of the version of SQL taught in most database classes.

Some database queries are simple: A company might, for instance, want a printout of the names and phone numbers of all of its customers. But it might also want a printout of the names and phone numbers of just those customers in a given zip code whose purchase totals exceeded some threshold amount over a particular time span. If each purchase has its own record in the database, the query will need to include code for summing up the purchase totals and comparing them to the threshold quantity.

What makes things even more complicated is that a database will generally store related data in different tables. For demonstration purposes, Bakke loaded several existing databases into his system. One of them, a database used at MIT to track research grants, has 35 separate tables; another, which records all the information in a university course catalogue, has 15.

Likewise, a company might store customers' names and contact information in one table, lists of their purchase orders in another, and the items constituting each purchase order in a third. A relatively simple query that pulls up the phone numbers of everyone who bought a particular product in a particular date range could require tracking data across all three tables.

Bakke and Karger's tool lets the user pull in individual columns from any table—say, name and phone number from the first, purchase orders and dates from the second, and products from the third. (The tool will automatically group the products associated with each purchase order together in a single spreadsheet "cell.")

A filter function just like that found in most spreadsheet programs can restrict the date range and limit the results to those that include a particular product. The user can then hide any unnecessary columns, and the report is complete.

### Hands-on approach

Previous academic projects have explored techniques for database query construction using editable flow-chart diagrams or virtual buttons that can be snapped together. But Bakke and Karger's tool enables what is known in computer science as "direct manipulation" of data.

"It really harkens back to our physical nature, that we're very comfortable with the idea that if I pick something up and I twist it, then it will twist, and if I shake it, it will shake" Karger says. "You want the same feeling when you're manipulating information in a computer—that you're picking up the information and pushing it this way or sliding it that way or cutting things out—instead of writing some instructions telling the computer to do something. And then the computer does it, and you say, 'Oh, that's not what I meant.'"

Bakke conducted two studies of the usability of his tool. In one of them, 14 participants were asked to construct a series of queries using the tool and then rated their experience using the System Usability Scale, a standard measure that allows the comparison of different types of software. The tool's scores put it at the 52nd percentile in the category of business software, which isn't bad for an academic research project. But the scores for Microsoft's Access database program are much worse—around the sixth percentile. "The way to describe that result is that database querying is hard, but we can make it tolerable," Bakke says.

At present, Bakke's tool enables query construction on an existing [database](#), but it doesn't enable the direct entry or modification of data. He expects to begin adding that functionality over the next six months, and his office wall is covered with a list of functions that he'd like to add and bugs he needs to repair. But his hope is to release the [tool](#) in a year or so.

"It's almost ironic," Karger says. "Eirik's software is far more robust than just about everything that graduate students have built. But he's not satisfied with releasing it in its current form. He's aiming for something of commercial quality."

"It turns out that when you're dealing with people's data, you really need to get it right," Bakke says.

**More information:** Expressive Query Construction through Direct Manipulation of Nested Relational Results. [DOI: 10.1145/2882903.2915210](https://doi.org/10.1145/2882903.2915210) , [http://people.csail.mit.edu/ebakke/research/sieuferd\\_sigmod2016.pdf](http://people.csail.mit.edu/ebakke/research/sieuferd_sigmod2016.pdf)

*This story is republished courtesy of MIT News ([web.mit.edu/newsoffice/](http://web.mit.edu/newsoffice/)), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

APA citation: New database interface looks like a spreadsheet (2016, July 11) retrieved 23 October 2019 from <https://techxplore.com/news/2016-07-database-interface-spreadsheet.html>

*This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.*