

# Interactive SQL Queries and Program Code in Presentations

Johannes Schildgen,<sup>1</sup> Florian Heinz<sup>1</sup>

**Abstract:** Nowadays, most database lectures are performed with an accompanying visual presentation that further illustrates the conveyed facts. Conventional presentation software allows dynamic elements up to a certain level, for example revealing or changing parts of the slide step by step, or even an interaction with the viewers by means of polls or similar mechanisms. Recently, HTML- and browser-based frameworks for presentations have emerged, which allow an even higher degree of flexibility due to the manifold possibilities of HTML5 and JavaScript. This paper presents an approach of how to interactively modify parts of a slide during the presentation, like SQL-based queries or program code snippets, and show the results pretty-printed on the corresponding slide in real-time. This enables the lecturer to easily show more examples, and answer and illustrate side questions, which they did not prepare in advance.

**Keywords:** Lecture Slides; SQL

## 1 Introduction

Today, a lecture is not thinkable without an accompanying presentation that helps to visually illustrate the topic. Often, this presentation is handed out to the audience and replaces, or at least supports, handwritten notices taken by the audience. Classical presentation software is, for example, Microsoft Powerpoint, Apple's Keynote, or LibreOffice Impress, which have roughly the same feature set available. Basic features are to reveal parts of a slide step-by-step or provide animations. To implement more advanced features for more flexibility in the presentation, software-specific plugins are needed here, for example, poll-plugins to interact with the audience.

Recently, another method to create presentations has become more and more popular: browser-based presentation frameworks that make heavy use of HTML5, CSS, and JavaScript. For example, one of such frameworks is reveal.js [EH13]. A big advantage is the comparatively easy extensibility of presentation features due to the open nature of the platforms used, to tailor the presentation to the specific needs of the lecture topic.

In this paper, we present an extension to reveal.js that can be used for lectures in computer science. Our first implementation was able to send SQL queries on the slides to a real database, executes them, and shows the results directly within the slides. Then, we extended our approach to supporting arbitrary program code.

---

<sup>1</sup> OTH Regensburg, Postfach 120327, 93025 Regensburg, Germany  
{johannes.schildgen,florian.heinz}@oth-regensburg.de

The traditional way of creating SQL slides is to think about an example, formulate a database query for it, execute it in a database system and copy both the query and its result into the presentation. The disadvantage of this approach becomes imminent when someone in the audience asks a *but what if?*-question, where the query is slightly altered. With the traditional tools, the lecturer has to stick with his prepared visualization and can only explain verbally, what will happen or they have to make a context switch and reproduce the query within an SQL client software.

This is where the new method described in this paper comes into play; a JavaScript-based plugin for reveal.js that executes a query shown in the presentation and presents the results to the audience. This enables the lecturer to flexibly react on questions regarding the queries that are raised by the audience. Further benefits of this feature are: No erroneous queries on the slides (it would show an error message), the avoidance of frequently switching between the presentation window and a console (and setting up an example environment), and consistent DB schemata and data values with less effort to produce, as they are based on a real relational database.

In programming lectures, frequently similar situations occur. Small changes to the code often result in completely different behavior. Operator precedence, lazy evaluation, out-of-order processing, and many things more can be illustrated swiftly if questions occur spontaneously. So, it is a big advantage if the code snippet is already part of a runnable program with the irrelevant parts hidden and its output presented on the slide: for one, there is no room for inconsistency or typos on the slide, because it is indeed compiled and executed. For another, there is no need for the lecturer to set up an IDE for the programming language in question and write boilerplate code to get a specific example running. This saves a lot of time and does not confuse the audience: The bigger context of the code part can be concealed, but modifications to the shown excerpt result in an immediate update of the output on the slide, together with possible compile-time and runtime errors.

## 2 Related Work

**reveal.js** The HTML5- and JavaScript-based framework reveal.js [EH13] is the foundation for the work presented here. Creating a presentation with reveal.js is similar to designing an HTML website. For example, each slide is one `<section>` element, and the sizes of a slide's contents are automatically adjusted to the size of the browser viewport. reveal.js offers a speakers view, math formulae, PDF export, syntax highlighting for code, slide transitions, and fragments, i.e., elements on a slide can be revealed step-by-step. One of the most useful features is the plugin interface<sup>2</sup>, which allows to extend the framework with self-programmed functions, as for example polls, diagram-creation functions, or dynamic SQL queries.

---

<sup>2</sup> <https://revealjs.com/plugins/>

**LaTeX Beamer** LaTeX Beamer is a popular alternative to PowerPoint and other presentation tools, especially in maths and science education. It works in a similar way as reveal.js: A text-based syntax is used to develop and style the content of the presentation slides, and then a PDF is generated, which can be presented to the audience with a PDF viewer. There are tools like SQLTeX [vE16] and LaTeXDB [Eß06], which act as a preprocessor for LaTeX files. They execute SQL queries that are part of the file on a relational database and write their results directly into the LaTeX document. This approach is similar to ours but—due to the output format PDF—it does not allow for interactively changing and re-executing the query while presenting the slides.

**Jupyter Notebook** Jupyter Notebooks [KI16, GG16, Pe18] are often used by data scientists to combine program code, documentation, and visualization within a file that can be inspected and executed within a browser. As this format is very interactive, users can change the program code and see its effects immediately. With plugins like sqlalchemy [Co08] or SQLFlow [Wa20], it is possible to include and execute SQL queries within a Jupyter Notebook. Another plugin, RISE [Av17], allows for presenting the notebook as a sequence of slides. While this approach follows the idea of creating an interactive document and presenting this in the form of slides, our approach is vice versa: We will create slides and enrich them with interactive elements.

**LiaScript** LiaScript [Di19] is a browser-based tool using an extended Markdown language to create interactive online courses. It is designed to be easy-to-use and to provide a high level of flexibility; the markdown source is rendered live in the browser itself. It also provides the ability to execute code snippets, perform quizzes, surveys and more. The focus is on providing an online course, while this work strives to augment a presentation with interactive elements, that is performed by a lecturer with a live audience.

### 3 Dynamic Content Evaluation in Presentations

A central element of a successful lecture is illustrating the theoretical concepts by means of examples. This helps to avoid misunderstandings and provokes questions from the public that otherwise would not have come up. Usually, the lecturer strives to find suitable examples and tries to anticipate possible questions to answer them in that course. This will, however, not be possible in all situations. So, a dynamic component is the best option to provide maximum flexibility during the presentation.

For example, the lecture comes to a point where NULL values in SQL are discussed and the lecturer tries to convey that each comparative relational operation, i.e. less than, greater, equality or inequality, with a NULL value is neither true, nor false, but NULL (“unknown”) in the result and the correct way to test for such values to use the IS NULL operation. Then, one slide might show a query `SELECT * FROM people WHERE email IS NULL` (see Figure 1).

# NULL Values

```
SELECT * FROM people WHERE email IS NULL
```

name	phone	email
James Smith	1-555-31337	-
Anna Doe	1-555-12325	-
Frank Hollywood	1-555-21312	-

Fig. 1: Dynamic Query

Below the SQL statement, an exemplary result table with several rows that might be a possible result of that query is shown. This is the primarily relevant information and will also be part of the (printed or digital) handout. But when dynamic queries are used, some more possibilities exist to further clarify this fact. The query is editable and the IS operator can be replaced by the relational equality operation =. The keyboard shortcut `ctrl` + `↵` re-executes that query live in an SQL database and shows that the result set is indeed empty. After that, the lecturer might be confronted with the question of what the result with the `!=` operation might be (people tend to think, that might yield all rows from the table then). Again, the operator is quickly replaced and the audience notices, that the result set is empty again. The reason for this can then be reiterated from the theoretical explanation beforehand and the chances for a deeper understanding of the underlying mechanisms rise.

Another often-heard question when talking about the JOIN operator is: “How would the result look with a left join?”. Using this framework, this can also be demonstrated easily by changing the query in the slide accordingly. These are rather simple examples, but basically all kinds of queries are possible.

The more complex the query, the more nuances can be demonstrated by changing small parts of the query and explaining the observed changes in the result. This is exactly what the SQL plugin is designed for. Figure 2 shows how the plugin works. For the database system in the backend, there are two fundamentally different possibilities. For one, some modern web browsers (e.g., Chrome and Opera) contain an internal WebSQL API that allows to create and query a relational database from JavaScript without the need of any external program or service. This also means, that the whole presentation can be held from local files without the need of a web server. This is convenient, however, there are some drawbacks with this approach. The Web SQL API has been deprecated since 2010. This leads to the fact, that Mozilla Firefox and other major players in the browser business do not support this API. The browsers that do support it all use SQLite as implementation, which provides a rich set of SQL features. However, sometimes more complex database operations have to be

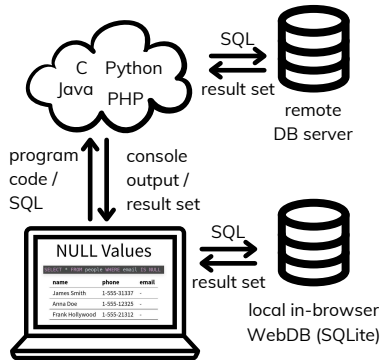


Fig. 2: Operating principle of the plugin

discussed, which might not be supported by SQLite, full outer joins and stored procedures for instance. For demonstrating code snippets in programming lectures, a backend server that compiles and executes code snippets is always needed.

To also be able to cover these and more topics within a database-systems lecture, this plugin also implements the possibility to connect to external database systems by proxying the query through a server-side PHP script, which constitutes the adapter between the dynamic JavaScript query functions and a database system, for example, a PostgreSQL server. Even NoSQL databases can be connected with some minor effort here. Either way, the user has the option to reset the database to a default state before loading the presentation or to stick with the current state, which might be preferable when methods for modifying the database scheme or the data itself are currently on-topic.

As already mentioned, the presentation framework used in this paper is reveal.js, which basically is a set of CSS and JavaScript files to set up a presentation by using HTML. The SQL query plugin uses the plugin interface of the framework and can easily be integrated by downloading our `sql.js` file (URL: see below) and adding it to the list of dependencies: `{ src: 'src/sql.js', async: true }` After that, dynamic queries can be defined in the presentation source like this:

```
<pre><code class="sql" contenteditable data-sql-engine="postgresql">
SELECT * FROM people WHERE email IS NULL</code></pre>
<span class="sqlresult"></span>
```

The result might be rendered as shown in Figure 1. If the query would contain an error, the message from the database would be presented to the audience in a popup window.

Within the `<code>` element, the following classes and attributes are supported by this implementation:

- `sql, java, ...`: classes for syntax highlighting from the library `highlight.js`; furthermore, an indicator for our plugin to execute the query or code
- `contenteditable`: makes the code editable and re-executable
- `data-sql-engine`: allows for using different backend database connections
- `dont_execute_sql`: The SQL query should not be executed (for `INSERT, UPDATE, ...`)
- `sqlresult`: class of `span` in which the result table will be shown
- `data-sql="some_id" / result="some_id"`: result table / program output will be shown in the `span` with the given id (if there are multiple code blocks on one slide)

The following will execute and show the results of a SQL query that is not shown on the slides: `<span data-sql-query="select * from tbl"></span>` This is useful for simply showing a full table, for showing results of relational-algebra expressions, or for faking queries (e.g., `A right join B` is shown on the slides, but `B left join A` is executed instead, because SQLite does not support right joins).

## 4 Conclusion

This paper presents a technique for creating slides for database lectures and other teaching presentations for computer science. SQL queries and program code are executed on a real system, the result is shown to the audience. This enables the lecturer to flexibly react to what-if questions from the audience without having to break the medium and frequently switch to a database client, an IDE, or similar.

While results of SQL queries are displayed in a table format, results of general program-code snippets is simple plain-text console output. By using JavaScript, this can be further processed to display, for example, charts, diagrams, chemical formulae, and much more.

The implementation of the dynamic SQL-query plugin for `reveal.js` can be found under <https://github.com/jschildgen/db-slides>.

## Description of the Demonstration

We will present the `reveal.js` presentation framework together with our extensions to the conference visitors in an interactive way. Everybody can click through slides, edit and execute SQL queries and code, and they can inspect the source code. Furthermore, we present more features and extensions of `reveal.js`, namely an ER-diagram creator and a poll plugin [Sc21]. The audience can use their own mobile phones to participate in polls and see how their voting will affect the presentation.

## Bibliography

- [Av17] Avila, Damian: Rise: Reveal.js-jupyter/ipython slideshow extension. 2017.
- [Co08] Copeland, Rick: Essential sqlalchemy. O'Reilly Media, Inc., 2008.
- [Di19] Dietrich, André: LiaScript: a domain-specific-language for interactive online courses. In: part of the Multi Conference on Computer Science and Information Systems 2019. p. 186, 2019.
- [EH13] El Hattab, Hakim: The HTML presentation framework. reveal.js, Jul 2013.
- [Eß06] Eßer, Hans-Georg: LaTeXDB - Integrates LaTeX and SQL databases. 2006.
- [GG16] Granger, B; Grout, J: JupyterLab: Building Blocks for Interactive Computing. Slides of presentation made at SciPy, 2016.
- [K116] Kluiver, Thomas; Ragan-Kelley, Benjamin; Pérez, Fernando; Granger, Brian E; Bussonnier, Matthias; Frederic, Jonathan; Kelley, Kyle; Hamrick, Jessica B; Grout, Jason; Corlay, Sylvain et al.: Jupyter Notebooks-a publishing format for reproducible computational workflows. In: ELPUB. 2016.
- [Pe18] Perkel, Jeffrey M: Why Jupyter is data scientists' computational notebook of choice. Nature, 2018.
- [Sc21] Schildgen, Johannes: Interaktive Vorlesungsfolien mit SQL-Unterstützung. Datenbank-Spektrum, 21(1):19–27, Mar 2021.
- [vE16] van Eijk, Oscar: SQLTex v2.0. 2016.
- [Wa20] Wang, Yi; Yang, Yang; Zhu, Weiguo; Wu, Yi; Yan, Xu; Liu, Yongfeng; Wang, Yu; Xie, Liang; Gao, Ziyao; Zhu, Wenjing et al.: SQLFlow: A Bridge between SQL and Machine Learning. arXiv preprint arXiv:2001.06846, 2020.