



Interaktive Vorlesungsfolien mit SQL-Unterstützung

Johannes Schildgen¹

Eingegangen: 28. September 2020 / Angenommen: 18. Dezember 2020 / Online publiziert: 21. Januar 2021
© Der/die Autor(en) 2021

Zusammenfassung

Mit PowerPoint oder LaTeX Beamer erstellte Vorlesungsfolien sind meist statisch und dienen hauptsächlich der Präsentation von Lehrinhalten. Als Alternative dazu werden drei Erweiterungen für das HTML- und JavaScript-basierte Präsentationsframework reveal.js vorgestellt, die für mehr Interaktion in der Datenbankenlehre sorgen sollen: (1) Eine Live-Ausführung von SQL-Anfragen und eine Darstellung des Anfrageergebnisses direkt auf der Folie; mit Möglichkeit zur Anpassung der Anfrage im Präsentationsbetrieb, (2) eine JSON-basierte Beschreibung von ER-Diagrammen, welche graphisch auf den Folien dargestellt werden sollen und (3) eingebettete Smartphone-Umfragen, um zwischendurch – ohne einen Kontextwechsel – Quiz-Fragen zu stellen.

Schlüsselwörter Datenbanken-Vorlesung · Präsentationen · Lehre

1 Motivation

In der Datenbankenlehre werden digitale und interaktive Lernelemente vor allem im Übungsbetrieb eingesetzt [17]. Das Hauptmedium in Vorlesungen – sowohl in Präsenz als auch in der Online-Lehre – sind jedoch computergestützte Folienvorträge [20]. Diese sind in der Regel mit einer Präsentationssoftware wie Microsoft PowerPoint oder Apple Keynote erstellt oder liegen in Form von LaTeX-Beamer-Dateien vor, welche in ein PDF-Format kompiliert werden, die schließlich zur Präsentation und gleichzeitig meist auch als Vorlesungsskript für die Studierenden dienen. Verglichen mit einem Tafelanschrieb oder der Verwendung von Overhead-Folien bieten computergestützte Präsentationen zwar deutlich mehr Möglichkeiten, jedoch wird von diesen de facto nur selten Gebrauch gemacht. Das Hauptargument für deren Einsatz scheint viel mehr der erhöhte Komfort für den Ersteller und Präsentator zu sein als ein tatsächlicher didaktischer Mehrwert. Eine typische PowerPoint-Präsen-

tation gleicht also aus Sicht der Studierenden größtenteils einer Overhead-Folien-Präsentation [25].

Ein besonders hoher didaktischer Mehrwert wird dagegen über einen kombinierten Einsatz verschiedener Lehrformen erreicht [2]. In einer Datenbankvorlesung kann dies beispielsweise das Zeichnen eines ER-Diagramms an der Tafel, die Ausführung einer SQL-Anfrage auf einer Datenbank oder das Live-Coding eines Java-Programms in einer Entwicklungsumgebung sein. Zusätzlich dazu werden auch vermehrt Browser-basierte Abstimmungssysteme eingesetzt, mit welchen Studierende an Quizzes und Umfragen teilnehmen können [5, 9]. Nichtsdestoweniger achten Lehrende darauf, sparsam mit einem damit verbunden Medienwechsel umzugehen, da ein zu häufiger Wechsel zwischen verschiedenen Fenstern (PowerPoint, SQL-Editor, IDE, Browser) schnell zu einer Verwirrung bei den Studierenden führen kann [8]. Es wird schwierig nachvollziehbar, was zusammengehört, und zu Hause bei der Nachbereitung werden Inhalte, die in den Folien fehlen – weil sie anderweitig präsentiert wurden –, leicht übersehen.

In diesem Artikel wird ein Ansatz präsentiert, welcher klassische Präsentationsfolien mit interaktiven Elementen wie dem Ausführen von SQL-Anfragen vereinigen soll. Der Ansatz verfolgt nicht das Ziel, alles Bisherige zu ersetzen. Stattdessen soll er als Ergänzung und zur Inspiration dienen und neuartige Möglichkeiten aufzeigen, um verschiedene Lehrmethoden miteinander verschwimmen zu lassen und dadurch ein stimmigeres Gesamtkonzept zu bilden. Hier

Die in diesem Artikel vorgestellten Plugins, eine Dokumentation sowie der Quellcode einer vollständigen Datenbanken-Vorlesung stehen auf <https://github.com/jschildgen/db-slides> unter der CC-BY-Lizenz frei zur Verfügung.

✉ Johannes Schildgen
johannes.schildgen@oth-regensburg.de

¹ OTH Regensburg, Regensburg, Deutschland

sind einige Beispiele, auf die in diesem Artikel noch detaillierter eingegangen wird:

- Eine SQL-Anfrage, die sich auf einer Folie befindet, wird an eine mit dem Präsentationstool verbundene Datenbank geschickt und dort ausgeführt. Während der Vorlesung kann die Anfrage modifiziert und erweitert werden; das Ergebnis aktualisiert sich dementsprechend.
- Auf einer Vorlesungsfolie erscheint eine Multiple-Choice-Frage und eine (immer gleiche) URL, über die die Studierenden ihre Antwort geben können. Nach einer Auflösung durch den Dozenten erscheint die Verteilung der gegebenen Rückmeldungen in Form eines Balkendiagramms.
- Der Dozent bewegt die Elemente eines auf einer Folie befindlichen ER-Diagramms gemeinsam mit den Studierenden, um unterschiedliche Modellierungsalternativen zu diskutieren.
- Code-Auszüge aus einem Java-Programm, die aus einer externen Datei stammen, werden auf der Folie eingeblendet.

Die genannten Beispiele verfolgen jeweils das Ziel, (1) den Wechsel zwischen Anwendungsfenstern zu reduzieren, (2) Elemente, die zusammengehören, gemeinsam darzustellen und damit für einen roten Faden zu sorgen, (3) die Lehrinhalte auch bei der Nacharbeitung zu Hause besser zu verstehen. Für die Dozentin oder den Dozenten gibt es zudem den Vorteil, dass die Vorbereitung an einer zentralen Stelle erfolgt, und zwar beim Erstellen der Folien. Dies verhindert auch, dass beispielsweise die Durchführung einer Umfrage vergessen wird.

Im nächsten Kapitel geht es um Präsentationstools im Allgemeinen und um das HTML-basierte Präsentationsframework `reveal.js`. In Kap. 3 werden drei Erweiterungen für `reveal.js` vorgestellt. Dabei handelt es sich um die oben dargestellten Funktionalitäten für SQL-Anfragen, Umfragen und ER-Diagramme innerhalb von Folien. Nach einem Vergleich mit verwandten Arbeiten in Kap. 4 folgt ein Fazit in Kap. 5.

2 Präsentationstools

2.1 PowerPoint, LaTeX Beamer und Co.

Der geläufigste Weg, Präsentationen zu erstellen, besteht daraus, in einer Software wie PowerPoint oder Keynote Text- und Grafik-Elemente auf Folien zu platzieren und diese Folien zum Zeitpunkt der Vorlesung auf einer Leinwand den Studierenden zu präsentieren. Als weitere gängige Tools sind Google Präsentationen und LibreOffice bzw. OpenOffice Impress zu nennen.

In den MINT-Fächern haben sich dagegen vermehrt LaTeX-basierte Präsentationen durchgesetzt. Unter Verwendung des LaTeX-Beamer-Pakets dienen Kommandos, die den Erstellern meist schon aus dem Anfertigen wissenschaftlicher Publikationen bekannt sind, zur Definition des Erscheinungsbilds der einzelnen Präsentationsfolien. Im Gegensatz zu PowerPoint wird hier jedoch kein What-You-See-Is-What-You-Get-Ansatz verfolgt. Stattdessen muss der LaTeX-Code erst in eine PDF-Datei umgewandelt werden, bevor diese in einem PDF-Anzeigeprogramm dem Publikum präsentiert werden kann.

HTML-basierte Präsentationsframeworks wie zum Beispiel `reveal.js`, `impress.js`, `Flowtime.js`, `Webslides` oder `deck.js` verfolgen einen ähnlichen Ansatz wie LaTeX. Auch hier wird zunächst textuell „programmiert“, was auf den einzelnen Folien zu sehen sein soll. Statt eines Kompilervorgangs ist zum Betrachten der Präsentation jedoch lediglich das Öffnen der HTML-Datei in einem Browser vonnöten. Ein Webserver wird dabei nur bei Verwendung komplexer Plugins wie dem im nächsten Kapitel präsentierten Umfrage-Plugin benötigt, für simple Präsentationen dagegen reicht ein Doppelklick auf die HTML-Datei. Die Frameworks selbst sind in der Programmiersprache JavaScript geschrieben. Sie ermöglichen ähnliche Funktionalitäten wie PowerPoint und LaTeX Beamer, beispielsweise Folienübergänge oder das schrittweise Einblenden einzelner Folienelemente.

Der Hauptvorteil von PowerPoint und Co. ist die einfache Bedienbarkeit und das schnelle Layouten mit graphischen Elementen. Um zwei mit einem Pfeil verbundene Kästchen an einer bestimmten Stelle auf der Folie zu platzieren sind nur wenige Mausklicks nötig. Dieses Vorhaben ist in LaTeX und HTML deutlich aufwändiger. Das TikZ-Paket in LaTeX bietet Kommandos zur Erstellung grafischer Elemente, in HTML kann SVG verwendet werden. Auch das Einbetten von Diagrammen und Graphen wird in LaTeX und HTML über entsprechende Pakete unterstützt. Oft wird hier aber der Einfachheit halber die benötigte Grafik in einem externen Programm (z. B. PowerPoint) erstellt und als Bilddatei in die LaTeX- oder HTML-Präsentation eingebunden. Dieses Vorgehen erschwert jedoch ein nachträgliches Editieren.

Als Vorteil von LaTeX-Präsentationen wird häufig die Darstellung von mathematischen Formeln und Gleichungen genannt. Zwar bieten auch PowerPoint und Co. einen Formeleditor, der auch eine Eingabe in LaTeX-Syntax erlaubt, jedoch ist hier vor allem bei häufiger Verwendung ein höherer Aufwand nötig als in LaTeX. Ein weiterer Vorteil von LaTeX ist die Darstellung von Programmcode mit Syntax-Highlighting. Ein gängiger aber etwas umständlicherer Ansatz, dies mit PowerPoint zu erreichen, ist das Einfügen von Code samt Formatierung, z. B. aus einer Entwicklungsumgebung oder einem Syntax-Highlighter-Programm. Da-

rüber hinaus ist LaTeX kostenlos, plattformunabhängig und es wird zum Betrachten der Präsentationen keine spezielle Software benötigt. Oft wird noch die Verträglichkeit mit Versionierungstools wie Git gelobt, was eine Kollaboration und Historisierung von Präsentationsdateien praktikabler macht als es bei Binärdateien der Fall ist.

Viele weitere Funktionen wie das Einbetten von Multimedia-Inhalten in Folien, Hintergründe, Verlinkungen, Foliennotizen oder eine Referentenansicht werden von allen Tools unterstützt.

Sowohl Programme der ersten Kategorie (PowerPoint, ...) als auch der zweiten (LaTeX) stoßen irgendwann an ihre Grenzen. Bei PowerPoint können nur die mitgelieferten oder über Plugins verfügbare Funktionen genutzt werden. Es existieren zwar einige PowerPoint-Erweiterungen zum direkten Einbetten von Umfragen (siehe Abschn. 4), das Ausführen von auf den Folien befindlichen SQL-Statements auf einer Datenbank ist jedoch fast undenkbar. In LaTeX dagegen stößt man immer dann an dessen Grenzen, wenn sich das Erwünschte nicht mit dem PDF-Format vereinbaren lässt. Ein schrittweises Erscheinen von Aufzählungspunkten wird beispielsweise in LaTeX Beamer damit gelöst, dass jedes neue Fragment eine neue Seite im PDF-Dokument bildet. Animationen mit Bewegungen jeglicher Art oder sich während des Präsentationsbetriebs dynamisch verändernde Elemente sind jedoch in einer PDF gar unmöglich.

Dadurch dass HTML-basierte Präsentationen mittels eines Webbrowsers angezeigt werden, haben sie ähnliche Vorteile wie LaTeX-Präsentationen: Sie sind plattformunabhängig, erfordern keine spezielle Präsentationssoftware und sind gut geeignet für eine Versionierung [22]. Gleichzeitig werden einige Nachteile von LaTeX eliminiert: Anders als PDF-Dateien unterstützt das HTML-Format vielerlei Animationen und Interaktionsmöglichkeiten. Mit HTML, CSS und JavaScript stößt man somit viel seltener an die Grenzen des Möglichen. Die wichtigsten Funktionen zur Erstellung von Präsentationen sind dabei im Präsentationsframework bereits eingebaut, für Spezielleres existieren Erweiterungen. Darüber hinaus können aber auch JavaScript-Bibliotheken oder Code-Schnipsel aus dem Internet oder anderen Projekten eingebunden werden, um eine bestimmte Aufgabe zu lösen. Möchte man beispielsweise chemische Strukturformeln auf einer Folie anzeigen, existieren entsprechende Bibliotheken bereits für die Website-Programmierung und können einfach wiederverwendet werden. Auch die Entwicklung eigener Funktionen und Erweiterungen ist nicht schwierig, wenn grundlegende JavaScript-Kenntnisse vorhanden sind.

2.2 reveal.js

Das Open-Source-Framework reveal.js [7] steht in Form einer Zip-Datei zum Download zur Verfügung, die alle nötigen HTML-, JavaScript- und CSS-Dateien enthält. Nach dem Entpacken bearbeitet man eine mitgelieferte Beispiel-HTML-Datei in einem Texteditor und betrachtet das Resultat schließlich im Browser. Eine solche HTML-Datei beinhaltet mehrere Sections und jedes `<section>`-Element entspricht dabei genau einer Folie. Innerhalb dieser können HTML-Tags wie `<h1>`, `<p>`, ``, ``, ``, usw. zum Einsatz kommen, um den Inhalt einer Folie zu definieren. Möchte man, dass ein Element erst nach einem Klick auf der Folie erscheint, vergibt man ihm das Attribut `class="fragment"`. Über weitere Klassen lässt sich das Verschwinden oder Animieren von Elementen regeln, aber auch komplette Folienübergangseffekte. Statt in HTML können die Folieninhalte auch in Markdown-Syntax formuliert werden, was das Layouten vor allem für HTML-unerfahrene Entwickler etwas einfacher machen kann. Mit gängigen HTML-Kommandos wie `<video>` oder `<iframe>` werden Videos, Web-Elemente und andere Multimedia-Inhalte in Folien eingebunden, Programmcode kommt in `<pre><code>`-Blöcke. Diese lassen sich über Attribute so konfigurieren, dass ein für die verwendete Programmiersprache passendes Syntax-Highlighting vorgenommen wird und Zeilennummern eingeblendet und bestimmte Zeilen farblich hervorgehoben werden. Mathematische Formeln lassen sich in reveal.js genau wie in LaTeX mittels `$... $` bzw. `...` einfügen. Erstere stehen dann im Fließtext, letztere bilden einen eigenen Absatz und können mehrzeilig sein.

Eine reveal.js-Präsentation lässt sich über die Drucken-Funktion des Webbrowsers in eine PDF-Datei exportieren, beispielsweise zur Erzeugung von Handouts. Entsprechend konfiguriert beinhalten diese auch Foliennotizen, welche in den einzelnen Folien innerhalb von `<aside class="notes">`-Tags verfasst werden können. Das Starten einer Präsentation im Vollbild erfolgt in reveal.js mit der Taste F. Der Kurzbefehl S startet eine Referentenansicht. Diese öffnet ein separates Browserfenster, in dem die aktuelle und die folgende Folie, sowie Foliennotizen und eine Uhr angezeigt werden.

2.3 Plugins für reveal.js

Im reveal.js-Repository auf GitHub [19] befindet sich eine Übersicht über meist von Dritten entwickelte Plugins, welche sich in Präsentationen einbinden lassen. Dazu zählen viele simple aber praktische Dinge, wie eine Zoom- oder Laserpointer-Funktion oder das Springen zu einer bestimmten Folie über die Tastatur. Über andere Plugins können Diagramme oder eine Word-Cloud angezeigt werden,

wieder andere dienen zur automatischen Generierung eines Inhaltsverzeichnis oder eines Navigationsmenüs.

Das Plugin *Sampler* verfolgt den Ansatz, Programmcode nicht direkt in die Folien zu schreiben, sondern dass dieser in externen Dateien verbleibt und dass lediglich auf diese verwiesen wird. Folgende Code-Zeile blendet beispielsweise die Zeilen 5 bis 8 einer Python-Datei ein: `<code data-sample="test.py#5-8"/>` Dieses Vorgehen verhindert zum einen Tippfehler, die beim direkten Verfassen von Code auf Folien passieren können, zum anderen hält es Folien bei Code-Anpassungen stets aktuell. Das Plugin *run-in-terminal* zeigt ebenfalls den Inhalt einer externen Datei an. Zusätzlich führt es diesen aber auch tatsächlich auf der Konsole aus und zeigt die Ausgabe des Programms direkt auf der Folie.

3 Erweiterungen für die Datenbanklehre

In diesem Abschnitt werden drei Plugins für reveal.js vorgestellt: Ein SQL-, ein ER-Diagramm- und ein Umfrage-Plugin. Die ersten beiden wurden speziell für eine Datenbank-Vorlesung entwickelt, letzteres eignet sich für beliebige Vorlesungen.

3.1 Live-Ausführung von SQL-Anfragen

Das *SQL*-Plugin für reveal.js erstellt bei seiner Initialisierung – also wenn die Präsentation geöffnet wird – mehrere Tabellen in einer SQLite-Datenbank, die im Browser gespeichert wird, und befüllt diese mit Beispieldaten. Es ist also kein Datenbankserver notwendig. Möchte man jedoch statt einer SQLite-Datenbank ein anderes Datenbankmanagementsystem einsetzen, ist es auch denkbar, über entsprechende Wrapper-Skripte in PHP, Java oder Node.js eine entsprechende Verbindung herzustellen.

Abb. 1 zeigt eine Folie zum `IS NULL`-Prädikat in SQL. Der HTML-Quellcode zu dieser Folie lautet wie folgt:

NULL-Werte

```
SELECT * FROM produkte WHERE hersteller IS NULL
```

produktnr	bezeichnung	preis	hersteller
88	Katzenfutter	4.99	-
91	Maschinenbau-Lehrbuch	22.9	-
92	Regal	100	-
998	Geschenverpackung	0	-
999	Katalog	0	-

Abb. 1 SQL-Plugin – Die gezeigte Anfrage wird ausgeführt und das Ergebnis darunter in Form einer Tabelle dargestellt

```
<section>
  <h2>NULL-Werte</h2>
  <pre><code class="hlsql"
    contenteditable>
    SELECT * FROM produkte
    WHERE hersteller IS NULL</code></pre>
  <span class="sqlresult"></span>
</section>
```

Es fällt auf, dass die auf der Folie dargestellte Tabelle, die das Anfrageergebnis zeigt, im Quellcode lediglich als `` auftaucht. Durch die Kennzeichnung `<code class="hlsql">` wird nicht nur das Syntax-Highlighting auf die Sprache SQL gesetzt, das SQL-Plugin erkennt dadurch auch den Inhalt dieses Elements als SQL-Anfrage und führt diese auf der SQLite-Datenbank aus. Das Ergebnis wird in Form einer Tabelle im `` eingeblenet. Soll eine Anfrage in modifizierter Form oder aber auch gar nicht ausgeführt werden, lässt sich dies über entsprechende Attribute konfigurieren.

Auch im Kapitel über relationale Algebra kann das SQL-Plugin eingesetzt werden. Hier dient es lediglich zur Darstellung einer Ergebnistabelle. Das spart Schreibarbeit und sorgt für eine konsistente Anzeige von Ergebnissen:

```
 $\pi_{\{kundennr, name\}}$ (Kunden) $
<span data-sql-query="select kundennr,
  name from kunden"></span>
```

Ein großer Vorteil des SQL-Plugins ist die Vermeidung fehlerhafter Anfragen. Ist eine Anfrage syntaktisch inkorrekt oder kommen nicht vorhandene Spalten- oder Tabellennamen darin vor, schlägt die Anfrage fehl und beim Anzeigen der Präsentation erscheint eine entsprechende Fehlermeldung. Als Resultat verbleiben nur solche Anfragen auf den Folien, die korrekt und deren Ergebnis zufriedenstellend sind.

Das im oben dargestellten Code-Beispiel verwendete Attribut `contenteditable` macht die Anfrage während des Präsentationsbetriebs im Browser editierbar. Mittels `Strg+Enter` wird sie erneut ausgeführt und das Ergebnis entsprechend aktualisiert. Auf diese Art und Weise kann man beispielsweise auf der gezeigten Folie live in der Vorlesung untersuchen, was passiert, wenn man `WHERE hersteller = NULL` statt `IS NULL` schreibt und es kann eine Diskussion angestoßen werden, warum denn das Ergebnis in diesem Fall leer ist. Auch zur Beantwortung von Fragen, die von Studierenden kommen, ist das interaktive Bearbeiten und Ausführen von SQL-Anfragen nützlich.

Natürlich ist es auch möglich, Anfragen zu kopieren und diese in einem separaten SQL-Editor wie PGAdmin oder DBeaver einzufügen. Dieses Vorgehen soll keinesfalls wegfallen und komplett durch das SQL-Plugin ersetzt werden. Stattdessen bietet sich das SQL-Plugin vor allem für simple Anfragen an und vermeidet einen zu häufigen Kontextwechsel. Für komplexere Anfragen und mehrschrittig erarbeitete Lösungen ist es weiterhin didaktisch sinnvoll, in einen SQL-Editor zu wechseln, da dies auch das Tool ist, in dem die Studierenden die Übungen lösen und mit dem sie auch später in der Praxis zu tun haben werden.

3.2 Darstellung von ER-Diagrammen

Einige Plugins für reveal.js verfolgen das Ziel, Diagramme, welche textuell innerhalb des Quelltextes beschrieben werden, graphisch auf den Folien darzustellen. Das *Chart*-Plugin dient beispielsweise zur Anzeige statischer sowie animierter Linien-, Balken-, Kuchendiagrammen. Das *Diagram*-Plugin stellt Graphen dar und lässt Knoten nach und nach animiert erscheinen. Auch für UML-Diagramme gibt es mehrere Plugins, sodass Klassendiagramme, Sequenzdiagramme etc. auf den Folien eingeblendet werden können. Unter Verwendung der *PlantUML*-Erweiterung wird ein im HTML-Quellcode beschriebenes UML-Diagramm beim Aufruf der Präsentation an einen PlantUML-Server geschickt, welcher ein PNG-Bild generiert und zurück an den Client schickt. Dies bedeutet jedoch, dass keine Animationen oder Interaktionen auf dem erzeugten Diagramm möglich sind.

Da PlantUML auch eine Notationsform für ER-Diagramme bietet, kann das Plugin für die Erstellung solcher zum Einsatz kommen. Der folgende Code generiert das in Abb. 2 gezeigte ER-Diagramm:

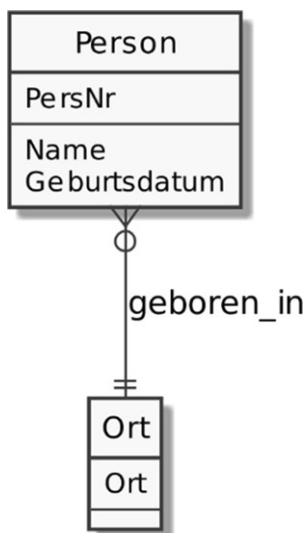


Abb. 2 PlantUML-Plugin – Das Plugin generiert UML-Diagramme sowie ER-Diagramme in Krähenfuß-Notation

```

<img style="width:6cm;"

uml="entity Person {
  PersNr
  -
  Name
  Geburtsdatum
}
entity Ort {
  Ort
  -
}
Person }o-|| Ort : geboren_in"/>
    
```

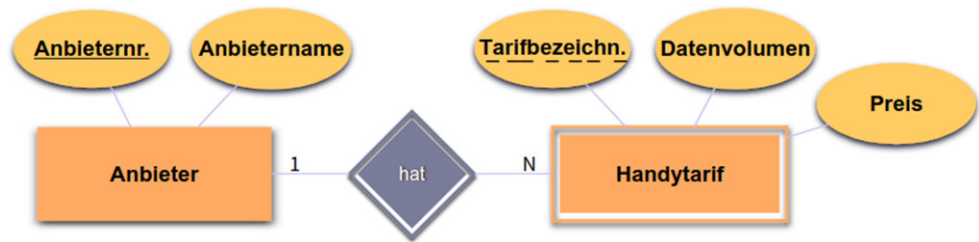
Wie in der Abbildung zu sehen ist, verwendet PlantUML die Krähenfuß-Notation; die Attribute werden im Entitätstypen-Kästchen positioniert. Ein horizontaler Strich separiert dabei zum Primärschlüssel gehörende Attribute von den Nichtschlüsselattributen.

Zur Darstellung von ER-Diagrammen in Chen-Notation wurde ein reveal.js-Plugin entwickelt, welches das Visualisierungsframework joint.js einsetzt. Da man Diagramme in joint.js anders als in PlantUML nicht textuell beschreibt sondern über die Erstellung von Objekten und über Methodenaufrufe, wurde eine JSON-basierte Syntax zur Beschreibung von ER-Diagrammen entwickelt. Das Plugin kreiert also aus dem im Folien-Quellcode beschriebenen JSON-String einen joint.js-Graphen und stellt diesen auf der Folie dar. Da es sich beim resultierenden Diagramm nicht um ein simples Bildobjekt sondern um eine Menge von SVG-Objekten handelt, lassen sich die einzelnen Objekte im Präsentationsbetrieb mit der Maus verschieben. Erweiterungen des Plugins sind denkbar, sodass auch das Hervorheben, Löschen, Ändern und Hinzufügen von Entitätstypen, Attributen und Beziehungen direkt im Browser ermöglicht wird. Das folgende Beispiel zeigt ein ER-Diagramm in JSON-Format und Abb. 3 die daraus generierte Darstellung:

```

[[
  { _e: "Anbieter", pos: [20, 80],
    attributes: [
      { _a:"Anbiaternr", pos: [5, 5],
        options:["primary"] },
      { _a:"Anbietername", pos: [120, 5] }
    ]
  },
  { _e: "Handytarif", pos: [350, 80],
    options: ["weak"],
    attributes: [
      { _a:"Tarifbezeichn.", pos: [300, 5],
        options:["extending_primary"] },
      { _a:"Datenvolumen", pos: [417, 5] },
    ]
  }
]]
    
```


Abb. 3 ER-Diagramm-Plugin – Darstellung des aus der JSON-Darstellung generierten ER-Diagramms



```

    { _a:"Preis", pos: [520, 40] }
    ]
    },
    [
    { _r: "hat", options: ["weak"],
      _e: ["Anbieter", "Handytarif"],
      card: ["1", "N"] }
    ]
    ]
    
```

Die JSON-Repräsentation eines ER-Diagramms besteht also aus einem Array der Entitätstypen und einem Array der Beziehungen. Deren Einträge haben jeweils einen Namen, einer Liste von Attributen und eventuelle Eigenschaften. Des Weiteren ist im gezeigten Beispiel zu jedem Element eine Position angegeben. Wird diese weggelassen, werden Elemente mehr oder weniger zufällig angeordnet. Dabei werden logisch zusammengehörige Elemente jedoch nah beieinander dargestellt. Bei der im Beispiel gezeigten Beziehung wurde keine Position angegeben, daher erscheint diese genau in der Mitte von den an der Beziehung teilnehmenden Entitätstypen. Verschiebt man die Elemente mit der Maus auf der Folie, erscheint im Browser auf der JavaScript-Konsole die neue Position, sodass diese einfach in den Quellcode übernommen werden kann. Ein Neuladen der Folie stellt die ursprüngliche Darstellung wieder her.

Das *ER-Diagramm-Plugin* unterstützt (schwache) Entitätstypen, (rekursive) Beziehungen, (zusammengesetzte, mehrwertige und abgeleitete) Attribute, (zusammengesetzte und erweiternde) Primärschlüssel und Generalisierung. Die genaue Verwendung kann der Dokumentation oder aus Beispielfolien entnommen werden. Beides befindet sich im GitHub-Repository (siehe Fußzeile zu Beginn des Artikels).

3.3 Eingebettete Smartphone-Umfragen

Abb. 4 zeigt die Verwendung des *Poll-Plugins*. Bei der Entwicklung dieses Plugins wurden die folgenden Ziele verfolgt: Eine Umfrage soll innerhalb der Folie eingebliendet werden und möglichst nicht die eigentlichen Folieninhalte überdecken, damit in einer Frage Bezug auf diese genommen werden kann. Des Weiteren sollen die Frage und die Antwortmöglichkeiten direkt in den Quellcode geschrieben werden können. Bei den meisten anderen Tools ist es immer erst nötig, die Fragen über ein Webinterface anzulegen. Und als letztes Ziel soll das Umfrage-Tool von den Studierenden und dem Lehrenden einfach zu bedienen sein. Dazu zählt auch, dass die URL zur Teilnahme immer gleich ist. Die Studierenden können ihr Smartphone angeschaltet auf dem Tisch liegen lassen und eine Umfrage erscheint immer genau dann, wenn sie auch auf einer Folie erscheint. Eine Umfrage wird wie folgt innerhalb einer Folie eingefügt:

Abb. 4 Poll-Plugin – Eine Frage erscheint, an der die Studierenden mit dem Smartphone teilnehmen können

NULL-Werte

```
SELECT * FROM produkte WHERE hersteller IS NULL
```

produktnr	bezeichnung	preis	hersteller
88	Katzenfutter	4.99	-
91	Maschinenbau-Lehrbuch	22.9	-
92	Regal	100	-
998	Geschenverpackung	0	-
999	Katalog	0	-

hersteller = NULL ist... 7

immer true

immer false

true, wenn hersteller NULL ist

false, wenn hersteller NULL ist

<https://eine-bestimmte-url>

```

<div class="poll fragment">
  <h1>hersteller = NULL ist...</h1>
  <ul>
    <li>immer true</li>
    <li data-poll="correct">immer
    false</li>
    <li>>true, wenn hersteller NULL
    ist</li>
    <li>>false, wenn hersteller NULL
    ist</li>
  </ul>
  <h2>https://eine-bestimmte-url</h2>
</div>

```

Während die Umfrage aktiv ist, zeigt eine Zahl oben rechts im Umfrage-Kasten die bisherige Anzahl an Rückmeldungen an. Die Auflösung erfolgt über einen Klick auf eine beliebige Antwort. Daraufhin wird die richtige Antwort fett markiert und die Verteilung der gegebenen Antworten in Form eines Balkendiagramms dargestellt. Mittels eines SQLite-Clients kann auf die Ergebnisse vergangener Umfragen zugegriffen werden, um beispielsweise semesterübergreifende Analysen durchzuführen.

Um das Umfrage-Plugin einsetzen zu können, wird ein öffentlich zugänglicher Webserver mit PHP-Unterstützung benötigt. Dessen URL wird in die Plugin-Einstellungen eingetragen und über diese URL kann an den Umfragen teilgenommen werden.

4 Verwandte Arbeiten

In diesem Kapitel werden Ansätze vorgestellt und verglichen, die ebenfalls eine Integration interaktiver Lehrformate in Präsentationen ermöglichen und die in diesem Artikel präsentierten Interaktionsformen zur Live-Ausführung von SQL-Anfragen sowie in Folien eingebettete Umfragen in ähnlicher Form bieten.

The integrated eBook [12] verfolgt auch den Ansatz, interaktive Lehrformate mit klassischen Formaten zu verbinden, jedoch nicht mittels Präsentationen sondern mittels eines Buches. Der Ansatz beschreibt, wie Multimedia-Elemente, Webseiten, Diskussionen und Umfragen in ein E-Book integriert werden können.

Vor allem in Data-Science-Anwendungen werden Jupyter Notebook [11] und JupyterLab [10] dazu genutzt, um Programmcode, dessen Ausgabe, Visualisierungen, interaktive Elemente und textuelle Beschreibungen in einem Dokument miteinander zu verbinden [15]. Eine Front-End-Komponente nimmt Programmcode und Benutzereingaben entgegen und sendet diese an ein Back-End, auf welchem der Code ausgeführt wird und welches das Ergebnis ans

Front-End zurückliefert, um dies schließlich dort darzustellen. Typischerweise kommen in Jupyter Notebook die Programmiersprachen Julia, Python und R zum Einsatz. Erweiterungen machen jedoch auch die Unterstützung für weitere Programmiersprachen möglich. Zur direkten Ausführung von SQL-Anfragen auf einer Datenbank innerhalb von Jupiter Notebook gibt es die Python-Bibliothek sqlalchemy [3]. Einen ähnlichen Ansatz verfolgt SQLFlow [23]. In beiden Tools werden SQL-Anfragen in ein Programm übersetzt, dessen Rückgabe direkt in Jupyter Notebook dargestellt wird. Damit ein Jupyter Notebook schließlich als Folien-Präsentation verwendet werden kann, bedarf es die Verwendung eines Slide-Plugins, beispielsweise RISE [1]. RISE fungiert als eine Art Sicht, die ausgewählte Blöcke des Jupyter Notebooks als Folien einer reveal.js-Präsentation erscheinen lassen. Die interaktiven Elemente bleiben dabei enthalten, sodass es während Präsentationsbetriebs möglich ist, Programmcode und SQL-Anfragen anzupassen und auszuführen. Das Ergebnis wird in Form eines Wertes, einer Tabelle oder eines Diagramms unter der Code-Eingabe dargestellt. Die in diesem Absatz präsentierten Werkzeuge ermöglichen eine Entwicklung eines interaktiven Jupyter Notebooks, welches schließlich in Form einer Folienpräsentation vorgeführt werden kann. Ein Vorteil ist neben der Mächtigkeit von Jupyter auch die Tatsache, dass das Notebook gleichzeitig als interaktives Vorlesungsskript für die Studierenden dienen kann. Der in diesem Artikel beschriebene Ansatz geht genau umgekehrt vor. Die Idee dahinter ist, dass zunächst eine klassische Folienpräsentation entwickelt und diese schließlich um interaktive Elemente wie die Ausführung von SQL-Anfragen erweitert wird.

SQLTex [6] und LaTeXDB sind Präprozessoren, um eine SQL-Ausführung in LaTeX-Dokumente und -Präsentationen zu bringen. Beide Programme arbeiten nach dem gleichen Prinzip. Sie übersetzen eine Eingabe-LaTeX-Datei, die SQL-Kommandos beinhaltet, in eine Ausgabedatei. In letzterer tauchen die Ergebnisse der jeweiligen Anfragen auf, beispielsweise innerhalb von Fließtext, als Liste oder in Form einer Tabelle. Bei fehlerhaften SQL-Anfragen schlägt der Übersetzungsprozess fehl oder zeigt eine Warnung. Dies sorgt ähnlich zu dem in diesem Artikel präsentierten Ansatz für eine Fehlerreduzierung in SQL-Anfragen und in deren Ergebnissen. Da jedoch SQLTex, LaTeXDB und auch LaTeX selbst statischer Natur sind, ist eine Interaktion innerhalb der Präsentation nicht möglich.

Zur Durchführung Smartphone-basierter Umfragen existieren viele Web-basierte Werkzeuge, beispielsweise Mentimeter [13] oder Pingo [18], welche im Hochschulbereich vermehrt eingesetzt werden [5]. Die meisten Tools haben jedoch einen Nachteil gegenüber dem im diesem Artikel präsentierten Umfrage-Plugin für reveal.js: Sowohl zur Erstellung als auch zur Durchführung von Umfragen muss in das Web-basierte Tool gewechselt werden. Die Umfragen sind

also kein Teil der Vorlesungsfolien selbst, sondern bestenfalls über einen Link in diesen miteinander verknüpft. Die Teilnahme an einer Umfrage durch die Studierenden erfolgt über eine URL, meist mit anschließender Eingabe eines Teilnahmecodes. In der Regel dauert es ein bis zwei Minuten, bis alle Teilnehmer verbunden sind. Um zumindest den Fensterwechsel zu vermeiden, kann ein PowerPoint-Plugin wie Web Viewer [24] verwendet werden. Mit diesem ist es möglich, beliebige Webseiten und damit auch Umfragen und deren Ergebnisse direkt in Folien einzubetten. Das Tool Poll Everywhere [16] lässt sich dagegen direkt nahtlos in Präsentationen integrieren. Das Programm bietet Plugins für PowerPoint, Keynote und Google Präsentationen, um dort direkt innerhalb einer Präsentation eine Folie einzufügen, die eine Frage, die Antwortmöglichkeiten, eine Teilnahme-URL und ein Balkendiagramm über die gegebenen Antworten enthält. Da diese Umfragen jedoch stets im Vollbild angezeigt werden, lässt sich eine Frage nicht gemeinsam mit anderen Folieninhalten darstellen. Das Einblenden von Bildern und Formeln innerhalb einer Umfrage ist jedoch möglich. Auch für andere Umfrage-Werkzeuge wie eduVote [4] und Mentimeter [13] existieren PowerPoint-Plugins, um Umfragen direkt in Folien einzufügen. Beim Mentimeter-Plugin ist es jedoch nicht möglich, Fragen direkt in PowerPoint zu erstellen. Stattdessen muss dies im Browser erfolgen, damit sie schließlich in eine PowerPoint-Folie über eine URL eingebettet werden können. Ein weiteres PowerPoint-Plugin ist SlideLizard [21]. Dieses Tool unterstützt jedoch nicht nur das Einfügen von Umfragen in Präsentationen, sondern bietet Teilnehmern auch die Möglichkeit, jederzeit Fragen oder Feedback am Smartphone zu verfassen, welches in der Referenzenansicht angezeigt wird. Zudem haben die Teilnehmer Zugriff auf alle Folien und können sich in diesen vor und zurück bewegen. Participoll [14] dagegen verfolgt einen minimalistischen Ansatz. Eine Frage und deren Antworten wird lediglich in Form von normalem Text auf der Folie platziert und nicht als interaktives Element. Participoll stellt auf den Endgeräten der Teilnehmer lediglich A/B/C/...-Buttons an und blendet das Ergebnis auf der Folie ein.

5 Fazit

Dieser Artikel zeigt diverse Möglichkeiten, wie interaktive Elemente in Vorlesungspräsentationen eingebettet werden können. Eine Live-Ausführung von SQL-Anfragen und die Darstellung von Umfragen direkt in den Folien vereinfacht nicht nur die Vorbereitung und vermeidet Fehler, es reduziert auch die Anzahl von Fenster- und Kontextwechseln. Nichtsdestoweniger ersetzen die hier gezeigten Methoden keine Live-Demos in einem SQL-Editor, sie ergänzen sie vielmehr. In jedem Falle steigern die gezeigten Methoden

jedoch die Interaktion innerhalb von Datenbanken-Vorlesungen, da durch sie typischerweise mehr SQL-Anfragen und Umfragen durchgeführt werden als es mit alternativen Tools gemacht würde. Auch die Einbettung von modifizierbaren ER-Diagrammen und Code-Beispielen steigert die Interaktion und regt zu Diskussionen an.

Die gezeigten Beispiele verwenden das Präsentationsframework reveal.js. Das in diesem Artikel vorgestellte Umfrage-Plugin soll in Zukunft aber auch in PowerPoint- und LaTeX-Beamer-Präsentationen eingebunden werden können.

In einer im Sommersemester 2020 durchgeführten Vorlesungsumfrage erhielten die beiden Vorlesungen, die reveal.js zusammen mit den vorgestellten Erweiterungen verwendeten, positive Kritiken. Die Studierenden lobten in Kommentaren vor allem die Ausführung von SQL in den Folien und die Umfragen. Eine umfangreiche Evaluation ist geplant, in der der didaktische Mehrwert für die Studierenden gezeigt werden soll.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Literatur

1. Avila D (2017) Rise: Reveal.js-jupyter/ipython slideshow extension
2. Ballstaedt SP (2009) Medien in der Hochschullehre: Medienbedarf, Visualisieren, Lernverhalten
3. Copeland R (2008) Essential sqlalchemy. O'Reilly Media, Sebastopol, USA
4. eduvote <https://eduvote.de>. Zugriffsdatum 28.09.2020
5. Eichhorn M (2016) Elektronische Abstimmungssysteme in der Hochschullehre-Empirische Untersuchung zu ersten Erfahrungen mit dem Audience Response System eduVote. DeLFI 2016–Die 14. E-Learning Fachtagung Informatik.
6. van Eijk O (2016) SQLTex v2.0
7. El Hattab H reveal.js: the HTML presentation framework. <https://revealjs.com>. Zugriffsdatum 28.09.2020
8. Faraday P, Sutcliffe A (1997) Designing effective multimedia presentations. In: Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, S 272–278

9. Fies C, Marshall J (2006) Classroom response systems: a review of the literature. *J Sci Educ Technol* 15:101–109
10. Granger B, Grout J (2016) Jupyterlab: building blocks for interactive computing. Slides of presentation made at sciPy
11. Kluyver T, Ragan-Kelley B, Pérez F, Granger BE, Bussonnier M, Frederic J, Kelley K, Hamrick JB, Grout J, Corlay S et al (2016) Jupyter notebooks—a publishing format for reproducible computational workflows. In: *ELPUB*
12. Mazza R (2008) The integrated ebook: the convergence of ebook, companion web site, and elearning. In: *Proceedings of the 2008 ACM workshop on research advances in large digital book repositories*, S 1–4
13. Mentimeter <https://www.mentimeter.com>. Zugriffsdatum 28.09.2020
14. Participoll <https://participoll.com>. Zugriffsdatum 28.09.2020
15. Perkel JM (2018) Why Jupyter is data scientists’ computational notebook of choice. *Nature* 563(7732):145
16. Poll everywhere <https://polleverywhere.com>. Zugriffsdatum 28.09.2020
17. Rakow TC, Faeskorn-Woyke H (2018) *Digitale Lehre im Fach Datenbanken*. GI-Edition, S 97
18. Reinhardt W, Sievers M, Magenheimer J, Kundisch D, Herrmann P, Beutner M, Zoyke A (2012) Pingo: peer instruction for very large groups. In: *European Conference on Technology Enhanced Learning*. Springer, Berlin, Heidelberg, New York
19. reveal.js – plugins, tools and hardware. <https://github.com/hakimel/reveal.js/wiki/Plugins,-Tools-and-Hardware>. Zugriffsdatum 28.09.2020
20. Sailer M, Schultz-Pernice F, Chernikova O, Sailer M, Fischer F (2018) *Digitale Bildung an bayerischen Hochschulen—Ausstattung, Strategie, Qualifizierung und Medieneinsatz*
21. Slidelizard <https://slidelizard.com>. Zugriffsdatum 28.09.2020
22. Thorogood M (2016) slidedeck.js: a platform for generating accessible and interactive web-based course content. In: *Proceedings of the 21st Western Canadian Conference on Computing Education*
23. Wang Y, Yang Y, Zhu W, Wu Y, Yan X, Liu Y, Wang Y, Xie L, Gao Z, Zhu W et al (2020) SQLflow: a bridge between SQL and machine learning (arXiv preprint arXiv:2001.06846)
24. Web viewer. <https://appsource.microsoft.com/en-us/product/office/WA104295828>. Zugriffsdatum 28.09.2020
25. Wedekind J (2007) *Digitale Medien in der Hochschullehre—Vision und Realität. Moderne Konzepte in der betrieblichen und universitären Aus- und Weiterbildung*