

# Semantik statt Syntax – Entwicklung einer Plug & Play Lernumgebung für Datenbankabfragen

Sven Jacobs,<sup>1</sup> Steffen Jaschke<sup>2</sup>

**Abstract:** Es wurde eine Webapplikation entwickelt, auf welcher Lernende SQL-Datenbankabfragen durch ein Zusammensetzen von SQL-Blöcken erstellen und ausführen können. So werden Syntaxfehler vermieden und somit mehr Zeit für die Korrektur von semantischen Fehlern geschaffen. Dabei können eigene Datenbanken verwendet werden, deren Datenbankstruktur visuell dargestellt wird. Über diese können einzelne Tabellen geöffnet werden. Nachdem fehlerhafte SQL-Datenbankabfragen ausgeführt wurden, erhalten Lernende die entsprechende Fehlernachricht und Fehlernummer, die ihnen bei der Korrektur behilflich sein sollen.

**Keywords:** SQL; Übungsumgebung; blockbasiert; visuell; Syntax

## 1 Einleitung

Der Lerninhalt Datenbankabfragen ist teils implizit, teils explizit in den Lehrplänen für das Fach Informatik in der Sekundarstufe I und II der meisten Bundesländer verankert [Ko18]. In der schulischen Praxis konnten zwei Hauptprobleme identifiziert werden. Zum einen werden professionelle Tools mit umfangreichem Funktionsumfang genutzt, was die Orientierung erschwert [UGB13], eine Einarbeitung vorab bedingt und meist die Installation eines Datenbankservers erfordert. Zum anderen befassen sich Lernende bei der Korrektur von Datenbankabfragen hauptsächlich mit Syntaxfehlern [Ah16], sodass weniger Zeit für die Berichtigung von Semantikfehlern verbleibt, wobei deren Lerneffekt als höher eingeschätzt wird [Ko18]. Um Syntaxfehler bei SQL-Datenbankabfragen zu minimieren, bietet sich ein Zugang über eine blockbasierte Lernumgebung an [GGM14]. Die Verwendung von blockbasierten Programmiersprachen führt im Vergleich zu textbasierter Programmierung zu einer erhöhten Motivation und einem höheren Lernerfolg [WW17], da Schüler\*innen diese Methode als einfacher bewerten. Begründet wird dies mit dem puzzleartigen Zusammensetzen, der Übersicht über die Bibliothek der verfügbaren Blöcke [WW15] und dem besseren Verhältnis zwischen „Werkzeug-Erlernzeit“ und „Werkzeug-Nutzzeit“ [MS16]. Da noch kein vergleichbarer blockbasierter Ansatz existiert, der beide Problemstellungen gleichzeitig adressiert, wurde die Webapplikation SqhELper<sup>3</sup> entwickelt, welche im Folgenden vorgestellt wird.

---

<sup>1</sup> Universität Siegen, Didaktik der Informatik, Hölderlinstraße 3, 57076 Siegen, sven.jacobs@student.uni-siegen.de

<sup>2</sup> Universität Siegen, Didaktik der Informatik, Hölderlinstraße 3, 57076 Siegen, steffen.jaschke@uni-siegen.de

<sup>3</sup> sqhelper.com

The screenshot displays the SQHeLpers user interface, which is divided into several sections:

- visual scheme:** Shows a database schema with tables: `projektmitarbeiter`, `projekt`, `mitarbeiter`, and `kind`. Arrows indicate relationships between these tables.
- textual scheme:** Shows the same database schema in a textual format, listing attributes and their data types for each table.
- Search block:** A search bar with a magnifying glass icon and a list of search categories: Statements, Tables and attributes, Operands and functions, Input, and Advanced.
- SQL query editor:** A drag-and-drop interface for building SQL queries. The current query is:
 

```
SELECT
  mitarbeiter.Name,
  mitarbeiter.MNr
FROM
  mitarbeiter
WHERE
  mitarbeiter.Geschlecht = 'w'
```
- Output area:** A text area showing the final SQL query, with a play button to execute it.

Abb. 1: SQHeLpers Nutzer\*innenoberfläche

## 2 Verwandte Arbeiten

Es existieren bereits webbasierte Lernumgebungen für SQL, in welchen eigens erstellte Aufgaben mittels textlicher Eingaben (z. B. eledSQL [UGB13]) oder per Drag & Drop (z. B. Sqool [Ko18]) bearbeitet werden können. Problematisch ist hierbei, dass die Eingaben per Text Syntaxfehler nicht beheben und die Eingabe per Drag & Drop auch die meisten semantischen Fehler ausschließt, sodass hier keine Lernprozesse durch das Korrigieren dieser stattfinden können. Darüber hinaus ist die Einstiegshürde zu den genannten Tools hoch, da Aufgaben und Datenbanken dort erst angelegt werden müssen und eine Registrierung erforderlich ist. Daneben existieren bereits einige blockbasierte Tools, um SQL-Datenbankabfragen zu erstellen (BlocklySQL [Pö19], SQLSnap [Mo15]). BlocklySQL ermöglicht keine Verwendung individueller Datenbanken und lässt inkorrekte Verbindungen zwischen Blöcken auf vertikaler Ebene zu. SQLSnap kann eine Verbindung zu extern gehosteten Datenbanken aufbauen, jedoch müssen Tabellen und Attributnamen textuell

eingegeben werden, sodass hier syntaktische Fehler nicht ausgeschlossen werden. Verglichen mit den hier vorgestellten Tools ermöglicht SQheLper die Verwendung eigener Datenbanken ohne Registrierung, da diese als MySQL-Dump hochgeladen werden können. Zudem wird das Erstellen von SQL-Datenbankabfragen mit 23 unterschiedlichen SQL-Blöcken durch die dynamische Darstellung der Datenbankstruktur unterstützt und die erstellte Abfrage in formatierter textueller Form ausgegeben.

### 3 Design

Nach Upload eines MySQL-Dumps werden die Lernenden zur Nutzer\*innenoberfläche von SQheLper weitergeleitet (siehe Abb. 1) und können im oberen Teil zwischen einer visuellen und einer textuellen Darstellung der Datenbankstruktur auswählen. Über einen Doppelklick auf die Tabellenüberschriften wird die jeweilige Tabelle geöffnet und deren Tupel angezeigt. Im mittleren Teil können Datenbankabfragen durch Zusammensetzen der SQL-Blöcke erstellt sowie Tabellen und Attributnamen per Dropdown ausgewählt werden, um Syntaxfehler zu vermeiden. Die SQL-Blöcke sind dabei fast gänzlich isomorph zur entsprechenden SQL-Syntax, sodass hier keine für blockbasierte Ansätze typische Vereinfachung stattfindet. [WW17] konnten zeigen, dass auch solche zur textbasierten Eingabe isomorphen Blöcke zu höheren Lernzuwächsen führen. Insgesamt stehen hier 23 verschiedene SQL-Blöcke zur Verfügung, welche folgende SQL-Syntax ermöglichen:

- Statements (FROM, WHERE, GROUP BY, ORDER BY, HAVING)
- Aggregatfunktionen (AVG, COUNT, MIN, MAX, SUM)
- Operanden (+, -, \*, ÷, =, ≠, ≥, >, ≤, <, LIKE, AND, OR, BETWEEN, IN)
- Inputs (Text, Nummern, Datum, Booleans)
- JOIN (INNER, LEFT, RIGHT, FULL)
- Unterabfragen

Es können nur Blöcke miteinander verknüpft werden, deren Verbund syntaktisch korrekt ist. Über Plus- und Minusbuttons können die SELECT-, FROM-, AND- und OR-Ausdrücke erweitert werden. Im unteren Teil der Nutzer\*innenoberfläche wird die aktuell erstellte Datenbankabfrage in formatierter Form dargestellt und die Lernenden können diese über den Play-Button ausführen. Ist die Abfrage fehlerhaft, so wird hier die entsprechende Fehlernachricht angezeigt, andernfalls wird die Ergebnistabelle ausgegeben.

### 4 Fazit

Da eigene Datenbanken ohne Registrierung verwendet werden können und sich die Webapplikation dynamisch an diese anpasst, ist die Einstiegshürde für die Verwendung im

Unterricht niedrig. Zudem wird durch den blockbasierten Ansatz auf textuelle Eingaben verzichtet, sodass Syntaxfehler minimiert werden. Im Zusammenspiel mit der übersichtlichen Nutzer\*innenoberfläche konnten erste Erprobungen zeigen, dass das Verhältnis zwischen „Werkzeug-Erlernzeit“ und „Werkzeug-Nutzzeit“ verbessert wurde.

## Literatur

- [Ah16] Ahadi, A.; Behbood, V.; Vihavainen, A.; Prior, J.; Lister, R.: Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and Its Application to Predicting Students' Success. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education. SIGCSE '16, Association for Computing Machinery, New York, S. 401–406, 2016.
- [GGM14] Gorman, J.; Gsell, S.; Mayfield, C.: Learning Relational Algebra by Snapping Blocks. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education. SIGCSE '14, Association for Computing Machinery, New York, S. 73–78, 2014.
- [Ko18] Koch, A.: SQooL.it – SQL per Drag&Drop lernen: Ein HTML5-Prototyp für die Sekundarstufe I, 2018, Stand: 13. 05. 2021.
- [Mo15] Modrow, E.: Neues von BYOB / Snap! LOG IN 35/1, S. 128–137, 2015.
- [MS16] Modrow, E.; Strecker, K.: Didaktik der Informatik. De Gruyter, Oldenbourg, 2016.
- [Pö19] Pöhner, N.; Schmidt, T.; Greubel, A.; Hennecke, M.; Ehmann, M.: BlocklySQL: A New Block-Based Editor for SQL. In: Proceedings of the 14th Workshop in Primary and Secondary Computing Education. WiPSCE'19, Association for Computing Machinery, Glasgow, Scotland, Uk, 2019.
- [UGB13] Ufert, A.; Grillenberger, A.; Brinda, T.: eledSQL - Entwicklung und Erprobung einer webbasierten Lernumgebung für Datenbanken und SQL. In (Breiter, A.; Rensing, C., Hrsg.): DeLFI 2013: Die 11. e-Learning Fachtagung Informatik. Gesellschaft für Informatik e.V, Bonn, S. 167–178, 2013.
- [WW15] Weintrop, D.; Wilensky, U.: To Block or Not to Block, That is the Question: Students' Perceptions of Blocks-Based Programming. In: Proceedings of the 14th International Conference on Interaction Design and Children. Association for Computing Machinery, Boston, Massachusetts, S. 199–208, 2015.
- [WW17] Weintrop, D.; Wilensky, U.: Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. ACM Transactions on Computing Education 18/1, 2017.