

ÜPS – Ein autorenfreundliches Trainingssystem für SQL-Anfragen

Marianus Ifland, Michael Jedich, Christian Schneider, Frank Puppe

Lehrstuhl für Informatik VI, Universität Würzburg
{ifland|jedich|c.schneider|puppe}@informatik.uni-wuerzburg.de

Abstract: SQL-Trainingssysteme haben in verschiedenen Studien ihre Nützlichkeit gezeigt. Ein breiter Einsatz hängt von einer Kombination aus einfachem Autorensystem für Dozenten mit attraktiver automatisch generierter Feedback-Struktur für Studierende ab sowie der Verwendung von einfach zu installierender Basissoftware ab. Das hier vorgestellte Übungs-Programm für SQL (ÜPS) kombiniert diese Merkmale und hat sich beim Einsatz in den Übungen einer großen Grundlagenvorlesung bewährt, bei der von ca. 300 Studierenden über 33000 SQL-Anfragen eingegeben und automatisch bewertet wurden.

1 Einleitung

Die Structured Query Language (SQL) ist eine standardisierte Datenbanksprache [DD97], mit der relationale Datenbanken definiert, bearbeitet und abgefragt werden können. Viele weit verbreitete Datenbank Management Systeme (DBMS), wie beispielsweise *MySQL*, *PostgreSQL*, *Microsoft SQL Server*, *IBM DB2* oder *Oracle Database* erfüllen den SQL Standard, wobei deren Abfragesprachen sogenannte SQL-Dialekte bilden. Im Informatik-Studium und verwandten Studiengängen sind relationale Datenbanken und somit auch die SQL ein wichtiges Thema, das nicht nur theoretisch behandelt, sondern auch praktisch eingeübt werden sollte. Nach Blooms Taxonomie [B156][AK01] liegen diese Lernziele also nicht nur in den Bereichen „Wissen“ und „Verstehen“, sondern vor allem im Bereich der „Anwendung“. Es ist daher wichtig, ausreichend viele Übungsmöglichkeiten anzubieten. Dafür eignen sich webbasierte Trainingssysteme sehr gut. Sie stellen Aufgaben zu einer Datenbank über eine Domäne, die mit der Eingabe einer SQL-Anfrage gelöst werden sollen. Die Studierenden erhalten dabei Rückmeldungen zu Syntax, Semantik und Ergebnis der eingegebenen Anfrage.

In der Literatur existieren bereits einige webbasierte Trainingssysteme wie *SQLT-Web* [Mi03], *SQLator* [Sa04], *SQL-ACME* [So06], *IDLE-SQL* [PK09] oder *eledSQL* [UBG13], die den Nutzen von SQL-Tutorsystemen überzeugend demonstriert haben. So würden 84 von 94 Studierenden das Trainingssystem *SQLT-Web* anderen Studierenden weiterempfehlen. Auch deutete sich ein erhöhter Lernerfolg jener Studierenden an, die intensiver mit dem System übten. Bei der Einführung von *IDLE-SQL* zeigte sich bei gleichbleibenden übrigen Faktoren eine signifikante Verbesserung der Ergebnisse in relevanten Klausuraufgaben gegenüber dem Vorjahr. Zudem stimmten mehr als 90% der Studierenden der Aussage zu, dass das Trainingssystem ein nützliches Lehr- und Lernwerkzeug sei. In einem Experiment mit Parallelgruppen schnitt die Gruppe, die das Tutorsystem *SQL-ACME* [So06] zur Vorbereitung nutzte, in einem Test besser ab als die Gruppe, die sich mit einem menschlichen Tutor vorbereitete. *eledSQL* ist speziell für

Schüler konzipiert und wurde in zwei Klassen der 9. Jahrgangsstufe mit 64 Schülern positiv aufgenommen.

Der Fokus von ÜPS liegt dabei auf einem sehr einfach bedienbaren Autorensystem, damit Dozenten ermutigt werden, möglichst viele Aufgaben zu stellen, wobei den Studierenden trotzdem aussagekräftiges Feedback auf verschiedenen Ebenen angeboten wird. Wir beschreiben im Folgenden ÜPS aus Sicht der Autoren und der Studierenden und zeigen dann in einer Evaluation, dass das Tutorsystem in den Übungen zu einer Grundlagen-Vorlesung für Studierende unterschiedlicher Studiengänge sehr gut angenommen wurde. Die verwendeten Technologien sind clientseitig HTML, JavaScript und CSS. Serverseitig wird Java7 auf Tomcat 7 mit den Frameworks *JavaServer Faces* und *Hibernate* verwendet. MySQL Server kommt sowohl als DBMS für die Szenarien-Datenbanken, als auch für die administrativen Datenbanken zum Einsatz. Durch die leichte Einbindbarkeit von ÜPS in eine Lernplattform wie Moodle mit einfachen Schnittstellen, benötigt ÜPS keine Nutzerverwaltung. ÜPS kann bei vorhandenem Tomcat Server mit MySQL Datenbank mit sehr wenig Aufwand installiert werden. Zusätzlich zum Kopieren der Anwendung im Dateisystem muss lediglich eine Konfigurationsdatei angepasst werden, in welcher beispielsweise Zugangsdaten zur Datenbank hinterlegt sind. Die administrative Datenbank wird beim ersten Start der Anwendung automatisch angelegt.

2 Autorensystem

Um Autoren zum Erstellen vieler Übungsaufgaben zu ermutigen, müssen Autoren nur folgende Eingaben tätigen:

- 1) Das Erstellen eines Datenbankszenarios mit externen Tools wie MySQL Workbench, HeidiSQL oder SQLyog oder direkt mit CREATE und INSERT Statements. Die Daten werden über ein Formular hochgeladen, wobei noch einige Meta-Daten wie Name und Beschreibung des Szenarios sowie ein ER- bzw. UML-Diagramm (als Bild-Datei) angegeben werden können. Die Beschreibung des Szenarios dient als narrativer Anker im Sinne der *Anchored Instruction* [Br90]. Dieser soll dazu dienen, Interesse zu wecken und Aufmerksamkeit auf das Wahrnehmen und Verstehen der gestellten Probleme zu lenken [Ni00].
- 2) Stellen beliebig vieler Aufgaben zu dem Szenario, die aus einem Aufgabentext und einer oder mehrerer Musterlösungen bestehen. Die Aufgaben beziehen sich dabei nur auf SQL-Anfragen, die für die Abfrage (SELECT) oder die Bearbeitung, also Ändern (UPDATE), Einfügen (INSERT) oder Löschen (DELETE), einer Datenbank verwendet werden. Aufgaben für Anfragen zur Definition einer Datenbank (z.B. CREATE) können in ÜPS nicht erstellt werden. Zu den Aufgaben kann ein Schwierigkeitsgrad (leicht, mittel, schwer) und eine Bewertungsart angegeben werden (Übungs- oder Prüfungsaufgabe, wobei bei letzteren das Feedback erst nach einer festgelegten Abgabefrist erscheint).

Abb. 1 zeigt das Web-Formular für die Erstellung von Übungsaufgaben. Als sehr hilfreich hat sich das Angebot der syntaktischen Prüfung der Musterlösung herausgestellt.

Aufgabe bearbeiten

✖
✖
Fehler in Query: SELECT price*stockAS revenue FROM books WHERE title='Animal Farm'
SQL Fehler: Die Spalte 'stockAS' existiert nicht!

Testfrage: Bestimmen Sie den potentiellen Umsatz des Buches "Animal Farm" (ohne Anführungszeichen). Hinweis: Als potentieller Umsatz wird hier das Produkt von Bestand und Preis bezeichnet. Nennen Sie diese Spalte "revenue" (ohne Anführungszeichen). [\[editieren\]](#)

Schwierigkeitsgrad: Anzahl Punkte: 2 [\[editieren\]](#)

Zuletzt bearbeitet: 16. Mai 2014 16:17 Uhr MESZ

Ursprung: Kopie von Übungsaufgabe mit ID: [15]

Übungsblatt: Select-Aufgaben, ID: 15, unbewertet.

Link: [Diese Aufgabe zum Testen in einem neuen Tab öffnen.](#)

Musterlösungen bearbeiten

SELECT stock*price AS revenue FROM books WHERE title='Animal Farm'	🗑	✎
SELECT price*stockAS revenue FROM books WHERE title='Animal Farm'	🗑	✎

+ Neue Lösung
↻ Syntaxprüfung
💾 Speichern
↶ Zurücksetzen
🗑 Verwerfen

Abbildung 1: Die Abbildung zeigt den bewusst sehr einfach gehaltenen Dialog zur Erstellung einer Aufgabe im ÜPS Autorensystem. Es wird eine Fehlermeldung angezeigt, da eine Syntaxprüfung durchgeführt wurde und die zweite Variante der Musterlösung einen syntaktischen Fehler enthält.

3 Tutorsystem

Die Studierenden können die Aufgabe über eine Web-Oberfläche bearbeiten (s. Abb. 2) und geben dazu SQL-Anfragen ein. Sie erhalten dann Feedback in bis zu drei Ebenen:

- erste Feedback Ebene: Falls die Anfrage syntaktisch nicht valide ist, wird die entsprechende von MySQL erzeugte englischsprachige Fehlermeldung angezeigt und ggfs. in deutscher Sprache erläutert.
- zweite Feedback Ebene: Syntaktisch valide Anfragen werden semantisch analysiert und mit der ähnlichsten Musterlösung verglichen, die mit der Levenshtein-Distanz ermittelt wird. Dabei wird aus den beiden Anfragen des Studierenden und der Musterlösung jeweils ein Parsebaum erstellt, anhand derer die Anfragen auf ihre Semantik hin untersucht werden. Der Parsebaum zerlegt die Anfragen hierarchisch in ihre Komponenten, so dass diese isoliert verglichen werden können. Unterschiede werden als Fehlermeldungen angezeigt. Dazu gehören fehlende Schlüsselworte wie (SELECT, UPDATE, INSERT oder DELETE), fehlende oder überflüssige Spalten in SELECT Anfragen oder in der WHERE Klausel sowie Fehler bei ORDER BY (bzw. GROUP BY) Klauseln.
- dritte Feedback Ebene: Die Anfrage und die Musterlösungs-Anfrage werden ausgeführt und die Ergebnisse verglichen. Unterschiede werden als Fehlermeldungen angezeigt. Sind die Ergebnisse strukturell und inhaltlich identisch, gilt die Aufgabe als gelöst. Zusätzlich können sich Studierende die Ergebnis-Tabellen anzeigen lassen und die Ergebnisse selbst vergleichen.

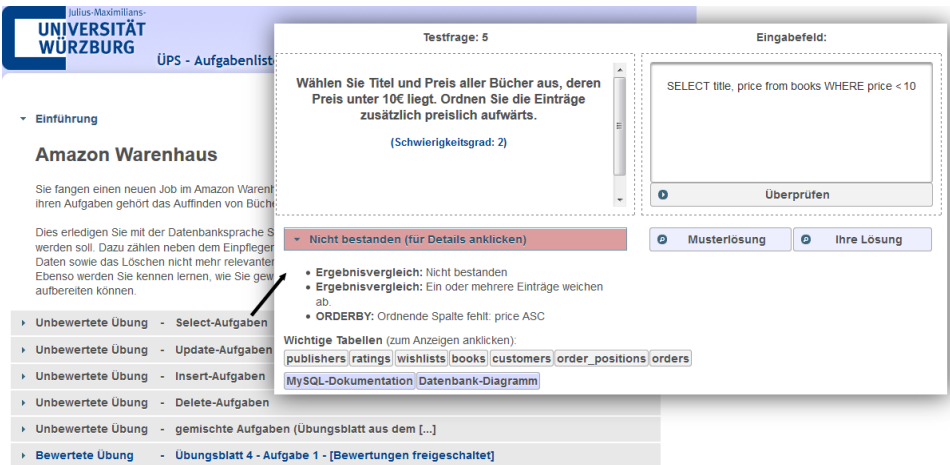


Abbildung 2: Die Abbildung zeigt links den Startdialog des Einführungs-Szenarios "Amazon Warenhaus" aus Sicht eines Studierenden. Zu verschiedenen SQL-Anfragetypen stehen Aufgaben zur Verfügung, nach deren Bearbeitung den Studierenden das Feedback angezeigt (rechts).

4 Evaluation

ÜPS wurde im Sommersemester 2013 in den Übungen zur Vorlesung Softwaretechnik im Datenbankteil mit Schwerpunkt auf SQL SELECT Aufgaben eingesetzt. Die Studierenden waren meist im ersten oder zweiten Semester und kamen aus unterschiedlichen Studiengängen: Informatik, Luft- und Raumfahrtinformatik, Wirtschaftsinformatik, Wirtschaftsmathematik, Mensch-Computer-Systeme und anderen Studiengängen. Insgesamt wurden 61 Aufgaben gestellt, davon 46 unbewertete, freiwillige Trainingsaufgaben zum Üben und 15 bewertete, verpflichtende Testaufgaben. Eine formale Evaluation mit Gruppe und Kontrollgruppe war nicht möglich, da aufgrund der Erfahrungen mit anderen SQL-Tutorsystemen zu erwarten war, dass die Studierenden, die ÜPS benutzen, einen Vorteil gegenüber denen haben würden, denen es vorenthalten würde. Wir haben folgende Variablen gemessen, die Indikatoren für die Akzeptanz und den Nutzen des Tutorsystems sind:

- 1) Eine umfassende Bearbeitung der freiwilligen Übungsaufgaben werten wir als ein Indiz, das die Studierenden das System nützlich finden. Tatsächlich haben ca. 95% der Studierenden (270 von 284) freiwillig zusätzliche Aufgaben bearbeitet. Insgesamt übersteigt die Anzahl der freiwilligen Bearbeitungen mit 20877 SQL-Anfragen die der verpflichtenden Bearbeitungen mit 12742 SQL-Anfragen deutlich (s. Tabelle 1).
- 2) Bei den Studierenden, die freiwillig Trainingsaufgaben bearbeitet haben, haben wir den Anteil der syntaktischen Fehler in den Trainingsaufgaben mit den Fehlern bei den späteren verpflichtenden Testaufgaben verglichen. Die Studierenden durften in den Testaufgaben bei syntaktischen Fehlern beliebig viele verschiedenen Lösungen eingeben ohne dafür Punktabzüge zu bekommen. Bei den SELECT Aufgaben

ist der Anteil der Bearbeitungen mit Syntaxfehlern von 63% bei den Trainingsaufgaben auf 39% bei den Testaufgaben gesunken, bei den UPDATE, INSERT oder DELETE Aufgaben sogar noch deutlicher. Die Bearbeitung von Trainingsaufgaben scheint also zu einer deutlichen Verringerung der Fehlerrate bei den Testaufgaben geführt zu haben.

- 3) Die späteren Zwischentestergebnisse der Studierenden, die freiwillige Übungsaufgaben bearbeitet haben, wurden mit denen der Studierenden verglichen, die diese nicht bearbeitet haben. Am Zwischentest haben 283 Studierende teilgenommen. Während die 262 Studierenden, die freiwillige Übungsaufgaben bearbeitet haben, einen Anteil bestandener Bearbeitungen von 79,5% hatten, kamen die 21 Studierenden, die dieses Angebot nicht wahrgenommen haben, nur auf 74,8% bestandene Bearbeitungen. Allerdings wissen wir nicht die Gründe, warum die 21 Studierenden das Angebot nicht wahrgenommen haben, ob sie z.B. besonders selbstbewusst oder faul waren.
- 4) Die Studierenden wurden in einem schriftlichen Fragebogen nach dem Zwischentest nach verschiedenen Aspekten des Trainingssystems und einer Gesamtbewertung der Konzeption und der Umsetzung gefragt. Dabei wurde das ÜPS in fast allen Aspekten mit Ausnahme der "Technik" durchschnittlich mit der Note „gut“ bewertet (s. Tabelle 2). Die Ausnahme kann daran liegen, dass es während des Bearbeitungszeitraums zu einem hardwarebedingten Ausfall des Servers kam und das System so für einige Stunden nicht erreichbar war. Der hohe Prozentsatz von 93% für den zukünftigen Einsatz elektronischer Tools und auch die Note 1,6 für das Konzept von ÜPS sowie das Bearbeiten freiwilliger Übungsaufgaben (Punkt 1) bestätigen die hohe Akzeptanz unter den Studierenden.

Aufgaben-Typ	SELECT	UPDATE	INSERT	DELETE	Gesamt
# Aufgaben	39/9	2/2	2/2	3/2	46/15
# Studierende	269/283	158/267	152/268	191/271	270/283
# Bearbeitungen	5998/2395	277/531	260/532	366/541	6901/3999
# SQL-Anfragen	18973/8715	547/1057	562/1311	795/1659	20877/12742
Anteil bestandener Bearbeitungen	74/72%	65/96%	46/86%	57/88%	71/79%
Ø Anzahl Anfragen pro Bearbeitung	3,2/3,6	2,0/2,0	2,2/2,5	2,2/3,1	3,0/3,2
Anteil Bearbeitungen mit korrekter Lösung bei der ersten Anfrage	22/46%	33/68%	7/54%	23/57%	22/51%
Ø Anzahl Anfragen bis zur korrekten Lösung (nur bestandene Bearbeitungen)	2,9/2,4	1,9/1,6	2,8/2,0	2,1/2,2	2,9/2,2

Tabelle 1: Bearbeitungsstatistiken von 46 freiwilligen Trainingsaufgaben und 15 verpflichtenden Testaufgaben (Notation: Trainingsaufgaben/Testaufgaben). Eine Bearbeitung bezieht sich auf einen Studierenden und eine Aufgabe, umfasst also mehrere SQL-Anfragen.

5. Diskussion und Ausblick

Die Ergebnisse zeigen, dass das ÜPS von den Studierenden viel genutzt wurde, eine gute Akzeptanz hatte und auch zu einer deutlichen Reduktion von Syntaxfehlern beim Vergleich von Trainings- zu Testaufgaben führte. Die Wirksamkeit des ÜPS für die SQL-Ausbildung hängt daher in erster Linie von einem großen Spektrum von interessanten

Übungsaufgaben ab. Da die Erstellung von Szenarien und Aufgaben sehr einfach ist, wollen wir in einem nächsten Schritt ÜPS mit diesem Peer-to-Peer-Ansatz von Studierenden für Studierende in der Praxis einsetzen.

Aspekt	Ø Note
Konzept des Tools, unabhängig von der Implementierung	1,60
Umsetzung des Tools, Gesamtnote	2,05
Aspekt Bedienbarkeit (Einarbeitung, intuitive Bedienbarkeit)	1,77
Aspekt Technik (Verfügbarkeit, Robustheit, Effizienz)	3,43
Aspekt Inhalt der Übungsaufgaben	1,83
Aspekt Fairness der Bewertung der Übungsaufgaben	1,94
Aspekt hilfreiche Erklärungen als Text oder nützliches Feedback	2,28

Tabelle 2: Bewertung von ÜPS durch 240 Studierende der Vorlesung mit Schulnoten von 1 bis 6. Zudem gaben 93% der Studierenden an, dass elektronische Tools (u.a. ÜPS) in Zukunft weiter eingesetzt werden sollen.

Literaturverzeichnis

- [AK01] Anderson, L. W.; Krathwohl, D. R.: A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives. 2001.
- [Bl56] Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; Krathwohl, D. R.: Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. 1956.
- [Br90] Bransford, J. D.; Sherwood, R. D.; Hasselbring, T. S.; Kinzer, C. K.; Williams, S. M.: Anchored instruction: Why we need it and how technology can help. In: Cognition, education, and multimedia: Exploring ideas in high technology, 1990; S. 115-141.
- [DD97] Date, C. J.; Darwen, H.: A Guide to the SQL Standard. Addison Wesley Pub Co Inc, 1997.
- [Mi03] Mitrovic, A.: An Intelligent SQL Tutor on the Web. In: International Journal of Artificial Intelligence in Education, Vol. 13, 2003; S. 173-197.
- [Ni00] Niegemann, H. M.: Die Konzeption von E-Learning: Wissenschaftliche Theorien, Modelle und Befunde. In: Kompendium multimediales Lernen, Springer-Verlag, Berlin, 2008; S. 17-40.
- [PK09] Pahl, C.; Kenny, C.: Interactive Correction and Recommendation for Computer Language Learning and Training. In: IEEE Transactions on Knowledge and Data Engineering, Vol 21, 2009. S. 854-866
- [So06] Soler, J.; Prados, F.; Boada, I.; Poch, J.: A Web-based tool for teaching and learning SQL. In: Proceedings of 7th International Conference on Information Technology Based Higher Education and Training (ITHET), 2006.
- [Sa04] Sadiq, S.; Orłowska, M.; Sadiq, W.; Lin, J.: SQLator: An Online SQL Learning Workbench. In: Proceedings of the Ninth Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE), 2004; S. 223-227.
- [UBG13] Ufert, A.; Brinda, T.; Grillenberger, A.: eledSQL – Entwicklung und Erprobung einer webbasierten Lernumgebung für Datenbanken und SQL. In: Proceedings of DeLFI 2013: Die 11. e-Learning Fachtagung Informatik, 2013; 167-178.