aus

Oliver J. Bott, Peter Fricke, Uta Priss, Michael Striewe (Hrsg.)

Automatisierte Bewertung in der Programmierausbildung

Digitale Medien in der Hochschullehre Band 6
2017, 420 Seiten, br., 42,90 €, ISBN 978-3-8309-3606-0

Waxmann Verlag GmbH
www.waxmann.com info@waxmann.com

4 Automatisierte Bewertung in der Ausbildung relationaler Datenbankabfragesprachen am Beispiel SQL

Oliver J. Bott, Felix Heine und Carsten Kleiner

Zusammenfassung
Der Beitrag führt ein in die Nutzung automatisierter Programmbewertung durch SQL-Grader in der SQL-Ausbildung und stellt die typischen Abläufe bei Aufgabenstellung und Bewertung vor. Weiterhin wird diskutiert, welche Lernziele mit dem Einsatz von SQL-Gradern erreicht werden können, und wie der Einsatz von SQL-Gradern gegen alternative Ansätze des rechnergestützten Assessments abzugrenzen ist. Der Hauptteil des Beitrags stellt potenzielle didaktische Einsatzszenarien anhand konkreter Beispiele im Rahmen eines Informatikstudiengangs und eines informatiknahen Studiengangs vor. Das Kapitel schließt mit einer Zusammenfassung didaktischer Einsatzszenarien und einem Ausblick auf offene Forschungs- und Entwicklungsfragestellungen.

4.1 Einleitung

Datenbanksysteme sind Basis zahlreicher Anwendungssysteme und werden in den unterschiedlichsten Anwendungsfeldern eingesetzt. Relationale und objektrelationale Datenbankmanagementsysteme (DBMS) dominieren seit Jahrzehnten

Das diesem Bericht zugrundeliegende Vorhaben eCULT (Teilvorhaben e-Assessment) wurde im Rahmen des Programms „Qualitätspakt Lehre“ vom Bundesministerium für Bildung und Forschung gefördert (Förderkennzeichen 01PL16066D).

den Markt. Die im Kontext dieser Systeme vorherrschende Datenbankabfragesprache Structured Query Language (SQL) [DD97] ist Gegenstand der Vermittlung in einer Vielzahl von Studienfächern und Lehrveranstaltungen.

Hierbei ist das Spektrum der Studienfächer nicht nur auf die Kerninformatik und die anwendungsbezogenen Informatikstudiengänge begrenzt, vielmehr spielt das Wissen um die Nutzung in relationalen Datenbanken abgelegter Daten in allen Disziplinen eine Rolle, in denen die Analyse und Weiterverarbeitung systematisch erhobener Daten relevant ist.

Die Vermittlung der SQL kombiniert üblicherweise die theoretische Vermittlung relationaler Datenbank- und Datenbankabfragekonzepte mit praktischen Anteilen in Form von Präsenzübungen und Übungen, die als Hausaufgaben bearbeitet werden. Zur Unterstützung der praktischen Übungen, aber auch zur Leistungsüberprüfung im Rahmen von formativen oder summativen Assessments oder auch zum Self-Assessment bietet sich die automatisierte Programmbewertung als ergänzendes didaktisches Instrument an. Damit einher geht die Zielsetzung, Studierende effizienter betreuen zu können und ihnen zudem zu mehr Selbstbestimmtheit beim Lernen zu verhelfen. Eine Vielzahl von Veröffentlichungen zu entsprechenden SQL-Trainingssystemen bzw. SQL-Gradern (s. auch Kapitel 12) konnte inzwischen den Nutzen derartiger Systeme für die Lehre demonstrieren, so zum Beispiel SQLT-Web [Mit03], SQLator [Sad+04], SQL-ACME [Sol+06], LEARN-SQL [Abe+08], IDLE-SQL [PK09], eledSQL [UGB13], aSQLg [Stö+13], XDaTA [Bha+15] oder ÜPS [Ifl+14b].

Für den Lehrenden stellt sich die Frage, wie diese Werkzeuge in der SQL-Ausbildung eingesetzt werden können. Dieser Beitrag stellt zur Beantwortung dieser Frage zunächst generelle Konzepte der Aufgabenstellung und Bewertung von SQL-Gradern vor, diskutiert mit diesen Systemen potenziell erreichbare Lernziele und grenzt den Einsatz automatisierter Bewertungssysteme für SQL gegen alternative e-Assessment-Ansätze ab. Anhand von Berichten aus der Praxis werden dann didaktische Konzepte der SQL-Ausbildung unter Einsatz automatisierter Programmbewertung vorgestellt. Das Kapitel schließt mit einem Fazit und einem Ausblick. Vorausgesetzt werden mindestens grundlegende Kenntnisse der SQL.

4.2 Aufgabenstellung und Bewertung

Eine Besonderheit in Bezug auf die Ausgestaltung von Aufgaben für die automatisierte Programmbewertung von SQL-Anweisungen gegenüber entsprechenden Konzepten für zum Beispiel objektorientierte Programmiersprachen ist, dass in der Regel

1. den Aufgaben ein vorgegebenes Datenbankschema zugrunde liegt, das
2. mit exemplarischen Daten gefüllt ist,
3. einzelne SQL-Anweisungen Gegenstand der Aufgaben sind und
4. eine Musterlösung in Form eines entsprechenden SQL-Statements vorliegt.

Die Bewertung der Aufgaben durch die Trainingssysteme erfolgt dann anhand einer Auswahl aus üblicherweise folgenden Bewertungskriterien:

1. Syntaktische Korrektheit der SQL-Anweisungen.
2. Einhaltung stilistischer Regeln (z. B. Großschreibung von Schlüsselworten) und/oder einschränkender Regeln zur Auswahl und Kombination von Schlüsselworten (z. B. zwingende Verwendung von INNER JOIN).
3. Vergleich der Ergebnisse der Ausführung der abgegebenen Lösung mit dem Ergebnis der Musterlösung in Bezug auf beispielsweise Spaltenanzahl und -benennungen sowie Anzahl, Sortierung und inhaltliche Übereinstimmung der enthaltenen Tupel.
4. Dauer bzw. allgemein Ressourcenbedarf der Bearbeitung der Anweisung.

Die Angabe einschränkender Regeln zur Auswahl und Kombination von Schlüsselworten ist dann erforderlich, wenn (wie häufig) verschiedene SQL-Anweisungen zum gleichen Resultat führen, aber nur eine vorgegebene Anweisung oder ein eingeschränktes Spektrum von Anweisungen für die Antwort zulässig sein soll. Hier sind Black-Lists verbotener oder White-Lists erlaubter Schlüsselworte gängige Konzepte, Trainingssysteme wie ÜPS [Ifl+14b] vergleichen zu diesem Zweck dagegen den Parsebaum des Statements mit Parsebäumen der Musterlösung(en) und bewerten Abweichungen als Fehler.

Wesentlicher einschränkender Faktor der SQL-Grader ist das Spektrum des unterstützten Sprachumfangs. Die meisten Systeme unterstützen ausschließlich auf SELECT basierende Datenbankabfragen als Teil der Data Manipulation Language (DML), andere erlauben auch DML-Anweisungen, die den Datenbankinhalt verändern, wenige auch Teile der Data Definition Language (DDL). Die Unterstützung von DDL- und auch die Datenbank verändernden DML-Anweisungen wird seltener von SQL-Gradern unterstützt, da die Bewertung deutlich komplexer ausfällt: Für jede Bewertung muss in der Regel eine Kopie der Datenbank bereitgestellt werden. SQL-Grader, die auf Basis eines konkreten DBMS arbeiten schränken den Sprachumfang in der Regel zusätzlich auf denjenigen des eingesetzten DBMS ein.

Die Rückmeldung der Trainingssysteme orientiert sich an den zur Verfügung stehenden Bewertungskriterien und liefert je nach Grader mehr oder weniger aussagekräftige Hinweise insbesondere im Falle negativer Bewertungen zu den genannten Kriterien. Zudem können die Bewertungskriterien mit einer Bewertungsvorschrift verknüpft sein, die im einfachsten Falle eine Punktzahl berechnet, die dann häufig zu einer Summenpunktzahl zusammengeführt wird. Auch werden von einigen Trainingssystemen die Musterlösung und/oder die Ergebnistabellen im Rahmen der Rückmeldung präsentiert.

Ein weiteres häufiges und didaktisch relevantes Merkmal der Trainingssysteme ist die Wiederholbarkeit einer Abgabe. Je nach Vorgabe des Lehrenden kann dem Lernenden in einem Testsystem ermöglicht werden, die Abgabe einer Lösung mehrfach oder beliebig oft zu wiederholen und so auf Basis der rückgemeldeten Hinweise eine fehlerhafte Lösung zu verbessern. Diese Funktionalität ermöglicht formative, das Lernen begleitende e-Assessment-Szenarien, in denen die Rückmeldungen des Systems der Lernunterstützung dienen. Für diagnostische e-Assessments zur Unterstützung der Selbsteinschätzung oder für summative e-Assessments zur abschließenden Prüfung und Bewertung der vom Lernenden erworbenen Kompetenzen des Lernenden spielt diese Funktionalität keine wesentliche Rolle.

Ein Autorensystem eines SQL-Trainingssystems bietet auf Basis dieser Grundkonzeption bei der Gestaltung von Aufgaben mehr oder weniger komfortabel folgende Optionen:

1. Anlage des zugrundeliegenden Datenbankschemas inklusive Datenbasis.
2. Textuelle Beschreibung der Aufgabenstellung.
3. Vorgabe einer Musterlösung.
4. Festlegung der Bewertung ggf. auf Basis der Parametrierung jedes einzelnen Bewertungskriteriums.
5. Vorgabe der Wiederholbarkeit der Abgabe einer Lösung gegebenenfalls unter Angabe zunehmend spezifischerer Lösungshinweise.

4.3 Unterstützbare Lernziele und Abgrenzung

Betrachtetet man die auf dieser Grundlage prinzipiell erreichbaren Lernziele, so sind durch den Einsatz von SQL-Trainingssystemen nach obigem Muster direkt

folgende Kompetenzen im Rahmen eines formativen oder summativen e-Assessments überprüfbar bzw. deren Vermittlung unterstützbar (unter Bezugnahme auf die Bloom'sche Taxonomie, vgl. [AKB01]):

1. *Kennen*: Lernende *kennen* die Syntax relevanter SQL-Kommandos und die typischen Regeln der stilistischen Gestaltung von SQL-Statements. Beispielfrage: „Formulieren Sie eine syntaktisch und stilistisch korrekte SQL-Anfrage zur Projektion der Tabelle Personen auf die Spalten Name und Vornamen."

2. *Verstehen*: Lernende *verstehen* die Semantik relevanter SQL-Kommandos. Beispielfrage: „Formulieren Sie eine SELECT-Anweisung zur Auswahl aller Patienten und der mit Ihnen in Behandlungsbeziehung stehenden ärztlichen Mitarbeiter. Verwenden Sie hierfür ‚INNER JOIN'."

3. *Anwenden*: Lernende können die SQL *anwenden*, um klassische Anwendungsprobleme zu lösen. Beispielfrage: „Formulieren Sie eine SQL-Anfrage zur Auswahl aller Patienten, die *nicht* in Behandlungsbeziehung zu irgendeinem ärztlichen Mitarbeiter stehen."

4. *Analysieren/Bewerten*: Lernende können verschiedene Lösungsvarianten im Hinblick auf deren Performance *analysieren* und die bestmögliche auswählen. Beispielfrage: „Formulieren Sie das folgende SQL-Statement mithilfe von JOINS und ohne Unterabfragen neu und überlegen Sie, welche Variante potenziell schneller ausgeführt werden kann." (das gegebene Statement enthält dabei eine Unterabfrage, die mit IN oder EXISTS eingebunden ist).

Die Lernzielklassen *Zusammenführen* und *Beurteilen* werden durch die automatisierte Programmbewertung in SQL höchstens marginal erreicht, da hier beispielsweise bei detaillierter Analyse der Ursachen für ein ineffizientes SQL-Statement weitere Werkzeuge zum Einsatz kommen sollten.

Insbesondere im Hinblick auf die Lernzielklassen 1 und 2 lassen sich die o. g. Lernziele auch mit alternativen Methoden des e-Assessments wie insbesondere die Verwendung von Lückentext-, Multiple-Choice- oder Zuordnungsaufgaben überprüfen bzw. unterstützen. Allerdings haben diese Optionen Nachteile gegenüber der automatischen Programmbewertung insbesondere im Hinblick auf die Anforderung an den Lernenden, das Antwort-Statement aktiv zu konstruieren. Lückentexte erzwingen zwar ein aktives Erarbeiten des gefragten SQL-Statements, lassen allerdings in der Regel nur ein eng umschriebenes Spektrum erlaubter Antworten zu und ermöglichen lediglich allgemeinere Hinweise zur Bewertung der Qualität der Lösung. Ob das Ergebnis auch mit einer eventuell von

der Musterlösung abweichenden, aber dennoch in Bezug auf das Ergebnis korrekten Lösung erreicht wird, kann bei einem Lückentext nur dann berücksichtigt werden, wenn alle möglichen Lösungen antizipiert und explizit beschrieben werden.

Bei der Verwendung von Multiple-Choice- oder Zuordnungsaufgaben besteht gegenüber der automatischen Bewertung der Nachteil, dass Lernende hierbei nicht gefordert sind, die Lösung mit ihren Bestandteilen aktiv zu erarbeiten, sondern *lediglich* eine vermutlich funktionierende Lösung zu erkennen. Ein weiterer Nachteil von Multiple-Choice- und Zuordnungsaufgaben gegenüber Aufgaben eines Testsystems eröffnet sich vor dem Hintergrund der Wiederholbarkeit einer Abgabe (s. o.): Während in formativen e-Assessment-Szenarien dem Lernenden in einem Testsystem nahezu beliebig oft ermöglicht werden kann, eine Lösung abzugeben und auf Basis der rückgemeldeten Hinweise eine fehlerhafte Lösung zu verbessern, erschöpft sich diese Möglichkeit bei Multiple-Choice und Zuordnungsaufgaben naturgemäß sehr rasch. Nichts desto trotz können insbesondere MC-Fragen zur korrekten Syntax oder Bedeutung von SQL-Anweisungen zum Beispiel im Rahmen von mit Audience-Response-Systemen (ARS; auch Klicker-Systeme) an das Plenum einer Vorlesung gestellten Fragen hilfreich sein, um Kennen und Verstehen von SQL-Anweisungen zu vertiefen oder den bisher erreichten Lehrerfolg zu überprüfen.

4.4 Einsatzszenarien und Erfahrungen

Im folgenden Kapitel werden anhand konkreter Einsatzszenarien und didaktischer Konzepte Optionen des Einsatzes automatisierter Programmbewertung in der SQL-Ausbildung diskutiert.

4.4.1 SQL im Studiengang Medizinisches Informationsmanagement

Im Bachelorstudiengang Medizinisches Informationsmanagement (BMI) an der Hochschule Hannover erfolgt im ersten Semester eine Einführung in die relationale Datenbanktheorie inklusive einer ersten Einführung in SQL. Im zweiten Semester werden in einer hierauf aufbauenden Veranstaltung die SQL-Kenntnisse vertieft. In beiden Veranstaltungen wird automatisierte Programmbewertung in unterschiedlichen didaktischen Szenarien eingesetzt.

4.4.1.1 Grundlagen relationaler Datenbanken (Datenbanken I)

Die Veranstaltung Grundlagen relationaler Datenbanken (Datenbanken I) führt im ersten Semester des BMI-Studiengangs in grundlegende Konzepte der Entwicklung relationaler Datenbanken ein:

- Grundbegriffe der Datenbanktechnologie (DBMS etc.),
- Datenbankentwicklungsprozess,
- Datenmodellierung auf Basis der ER-Modellierung,
- Grundlagen des relationalen Modells (insb. Einführung in die relationale Algebra) und Abbildung von ER-Modellen in Relationenschemata,
- Einführung in die SQL (DDL und DML).

Die Veranstaltung ist organisiert in eine zweistündige Vorlesungseinheit im Plenum und eine vierstündige praktische Übung in Kleingruppen. In der Übung wird anhand eines konkreten DBMS die Datenbankentwicklung eingeübt. In die Vorlesung integriert sind Übungseinheiten, die partiell in Form begleitender e-Assessment-Aufgaben unter Einsatz eines ARS im Präsenzunterricht durchgeführt werden, sowie Aufgaben, die freiwillig als Hausaufgabe bearbeitet werden können. Für die Lehrveranstaltung wird ein begleitender Kurs im Lernmanagementsystem (LMS) Moodle angeboten, über den Materialien wie Folien, Übungsaufgaben und weitere Materialien zugänglich gemacht werden.

Automatisierte Programmbewertung wird auf Grundlage des Graders aSQLg (s. Kapitel 12) integriert in das LMS Moodle im letzten Viertel der Veranstaltung durch Angebot von zwei Übungsblättern mit jeweils 10 Aufgaben (s. Abb. 4.1) eingesetzt. Bewertet wurden Syntax, Kosten (Zeit) und Korrektheit des Ergebnisses (s. Abb. 4.2). Auf Grundlage eines durchgängig in der Lehrveranstaltung verwendeten Datenmodells einer Hochschule werden SQL-Aufgaben in Bezug auf SELECT-Aufgaben gestellt. Dabei steigert sich der Schwierigkeitsgrad der Aufgaben im ersten Aufgabenblatt von einfachen Projektionen und Umbenennungen über die Erzeugung sortierter Ergebnisrelationen und die Ausblendung doppelter Ergebnisse hin zu Selektionen mit einfachen und komplexeren Selektionsprädikaten. Das zweite Aufgabenblatt fokussierte dann auf Kreuzprodukte und Kreuzprodukte mit Selektionsprädikaten im Hinblick auf die Nutzung von Beziehungen zwischen Entitäten, Natural Joins und Theta-Joins, Mengenoperatoren, Aggregationsoperatoren und Gruppierungen ohne und mit Filterung über

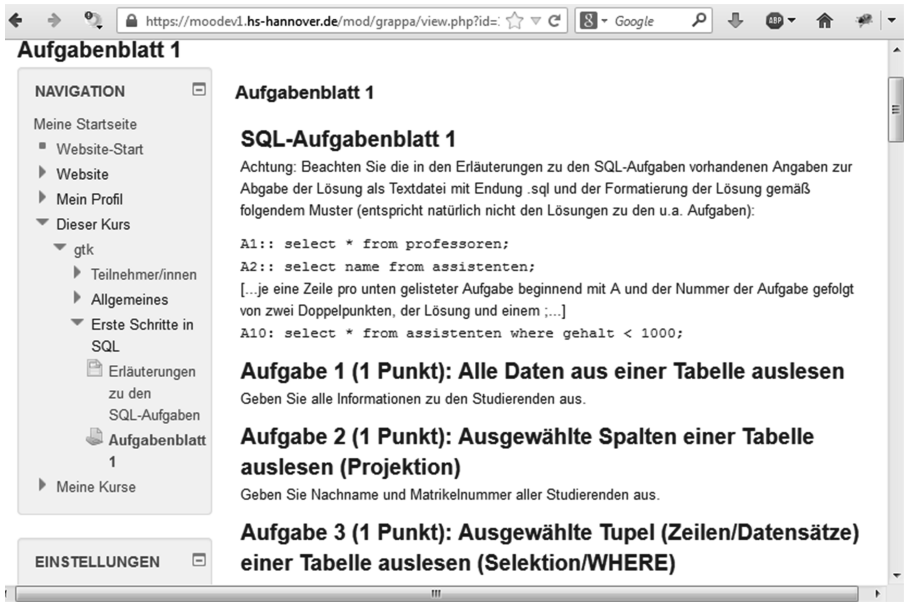

Abbildung 4.1: Beispielhafte Aufgabenbeschreibung eines SQL-Übungsblatts

HAVING sowie äußere Joins zur Auswertung von (Nicht-)Beziehungen. Dabei wurden so oft wie möglich Aufgaben gestellt, die bereits in zwei analogen Aufgabenblättern zur relationalen Algebra gestellt wurden – in diesem Fall allerdings ohne Option der automatisierten Programmbewertung. Bei Evaluationen dieses Einsatzszenarios im Wintersemester 14/15 (n=44 Studierende) und im Wintersemester 15/16 (n=43) haben 77 % (n=43) bzw. 64 % (n=32) der Studierenden das freiwillig nutzbare Angebot in Anspruch genommen. Die Übersichtlichkeit der Oberfläche sowie die Qualität des Feedbacks insgesamt wurde mit gut bewertet (jeweils Schulnote 2,1 (n=35/34) bei Noten von 1-5 bzw. Note 2,0 und 2,2 (n=30)) und die Geschwindigkeit mit gut bis sehr gut (1,6 (n=34) bzw. 1,9 (n=30)). 12 % (n=33) bzw. 3 % (n=29) der Studierenden waren eher nicht der Meinung, dass der Einsatz des Programms geholfen hätte, Fehler im Quellcode der SQL-Lösung zu finden, 70 % bzw. 60 % bewerteten diese Frage mit „eher Ja“ bzw. „Ja“, der Rest mit „teils-teils“. 87,5 % (n=32) bzw. 79,3 % (n=29) der Studierenden können sich vorstellen, automatisierte Programmbewertung auch in anderen Fächern einzusetzen. Die Didaktik der Lehrveranstaltung insgesamt wurde mit 1,5 (n=24) und 1,9 (n=10) bewertet, wobei die letzte Bewertung aufgrund der niedrigen Teilnehmerzahl nur bedingt aussagekräftig ist. Ergänzend angemerkt sei, dass der Einsatz des ARS, eingeführt im Wintersemester 13/14, im Wintersemester 14/15 mit der Note 1,2 (n=47) als sehr gut benotet wurde. Das ARS wurde allerdings nicht im Kon-

text SQL, sondern im Zusammenhang insbesondere mit der ER-Modellierung, der Überführung von ER-Modellen in Relationenschemata und der relationalen Algebra eingesetzt, würde sich aber natürlich analog für die SQL-Vorlesungsanteile eignen.

4.4.1.2 Vertiefungsveranstaltung Datenbanken II

Die Veranstaltung Datenbanken II des BMI-Studiengangs vertieft die im ersten Semester vermittelten SQL-Grundkenntnisse. Vermittelt werden alle relevanten Sprachbestandteile der DDL, DML und der DQL. Weiterführende Themen sind Transaktionssicherheit, Stored Procedures/Trigger und datenbankbasierte Anwendungsentwicklung.

Die Veranstaltung ist organisiert in eine zweistündige Vorlesungseinheit im Plenum und eine zweistündige begleitende Übung in Kleingruppen. Für die Lehrveranstaltung wird analog zu Datenbanken I ein begleitender Kurs im Lernmanagementsystem (LMS) Moodle angeboten. In der Übung wird parallel zu den Vorlesungsinhalten der jeweils behandelte Sprachumfang eingeübt. Da im Rahmen der Übungen zur Vorlesung bereits Aufgaben zu SQL gestellt und die Lösungen diskutiert werden, beschränkt sich die Nutzung der automatisierten Programm-

A10 2.0/2.0

Prüfungstyp	Punkte	Kommentare
Syntaxprüfung	0.67/0.67	Die Syntax ist korrekt.
Kostenprüfung	0.67/0.67	Kosten Ihres Statements: 4.0. Absolute Kostenobergrenze ist: 100000. Kosten der Musterlösung: 4.0. Die Kosten der Abfrage sollte kleiner als 8.0 sein. Die Kosten der Abfrage sind ok.
Ergebnisprüfung	0.67/0.67	Die Spaltenanzahl Ihres Ergebnisses ist richtig. Die Datentypen der Spalten Ihres Ergebnisses sind richtig. Die Namen der Spalten Ihres Ergebnisses sind richtig. Ihr Ergebnis hat die richtige Zeilenanzahl. Die Werte Ihrer Ergebniszeilen sind richtig. Die Sortierung der Ergebnisse ist richtig. Das Ergebnis ist vollständig richtig.

Abbildung 4.2: Beispielhaftes Feedback des aSQLg-Graders

bewertung – erneut wird aSQLg integriert in Moodle eingesetzt – auf den Zeitraum der Klausurvorbereitung. Bereitgestellt wird eine Probeklausur mit 20 Aufgaben zu SQL, davon sieben Aufgaben zu Datenbankabfragen mit dem SELECT-Statement. Das Spektrum der Aufgaben umfasste komplexere Abfragen über eine und mehrere Tabellen inklusive Sortierung und Selektion, Aggregationen und Gruppierungen und Unterabfragen. Nur diese Aufgabenkategorien konnten mit aSQLg automatisiert bewertet werden, die weiteren aufgrund des diesbezüglich eingeschränkten Sprachumfangs des verwendeten Graders noch nicht.
Insgesamt nutzten im Sommersemester 15 (n=39) 31 % der Studierenden (n=32) das freiwillige Angebot. Die Übersichtlichkeit wurde ebenso wie die Geschwindigkeit der Rückmeldung mit Note 2,4 (n=20, n=12) bewertet, die Qualität des Feedbacks im Hinblick auf dessen Unterstützungspotenzial die richtige Lösung zu finden mit 2,5 (n=10). 27,3 % (n=11) empfanden das Feedback geeignet, um Fehler in der eigenen Lösung zu finden, 27,3 % waren diesbezüglich unentschieden, 45,5 % schätzten das Feedback als diesbezüglich wenig hilfreich ein. Bei vergleichsweise geringer Antwortrate ist dieses im Verhältnis zu den Evaluationsergebnissen des Wintersemesters 14/15 unerwartete Ergebnis der Bewertung der Nützlichkeit des Feedbacks mit Vorsicht zu interpretieren. In den Freitextantworten nannten die Studierenden technische Probleme des Systems während der Vorbereitungszeit, was sich negativ auf die Bewertung des Feedbacks ausgewirkt haben könnte. Die Studierenden, die das System nutzen konnten, gaben in den Freitextrückmeldungen zum Teil positive Rückmeldungen, zum Teil wünschten sie sich detailliertere Informationen zu den eigenen Fehlern. Dies untermauert die Vermutung, dass gerade bei komplexeren Aufgaben die auf Syntax, Ausführungszeit und Ergebniskorrektheit fokussierten Rückmeldungen des Systems nur noch sehr bedingt geeignet sind, die Gründe der Fehler in der eigenen Lösung zu finden. 83,3 % (n=18) der Studierenden können sich vorstellen, automatisierte Programmbewertung auch in anderen Fächern einzusetzen.

4.4.2 SQL im Bachelorstudiengang Angewandte Informatik

Der Bachelorstudiengang Angewandte Informatik (BIN) an der Hochschule Hannover ist ein sechssemestriger Informatikstudiengang. Im Bereich der Datenbanksysteme sieht das Curriculum drei Module im Pflichtbereich vor:

- eine Grundlagenveranstaltung (Datenbanken), die sich mit konzeptioneller Datenmodellierung, Grundlagen relationaler Datenbanken sowie einer Einführung in SQL befasst,

- eine Vertiefungsveranstaltung (Informationssysteme I), die Interna relationaler Datenbanken, Anfrageoptimierung, Transaktionen und eine Vertiefung in SQL zum Inhalt hat sowie

- eine weitere Vertiefungsveranstaltung (Informationssysteme II), die sich mit der Anwendungsprogrammierung von relationalen Datenbanken sowie weiteren Datenbankkonzepten beschäftigt.

Für die automatisierte Bewertung von SQL sind somit insbesondere die ersten beiden Veranstaltungen relevant.

4.4.2.1 Datenbanken

Insbesondere in der Basisveranstaltung, in der die Studierenden erstmals mit SQL in Kontakt kommen, ist eine automatisierte Bewertung von SQL sinnvoll, da hier einerseits die größten Teilnehmerzahlen zu erwarten sind. Andererseits sind die zu erwartenden Fehler der Studierenden leichter automatisiert zu erkennen, da es sich eher um grundlegende Probleme handelt. Daher kann hier die Lernunterstützung besonders effizient eingesetzt werden. Allerdings spielen in dieser Veranstaltung neben einfachen Anfragen besonders auch Anweisungen der DDL und DML eine große Rolle. Diese werden bisher jedoch noch nicht vom eingesetzten Grader aSQLg unterstützt (s. Kap. 12.7). Aus diesem und anderen organisatorischen Gründen konnte an der Hochschule Hannover bisher jedoch noch keine systematische Auswertung des Einsatzes von aSQLg in der Veranstaltung Datenbanken erfolgen.

4.4.2.2 Informationssysteme I

In dieser Veranstaltung stehen im SQL-Teil insbesondere komplexere SELECT-Abfragen im Vordergrund, die sich ebenfalls sehr gut für eine automatisierte Bewertung eignen. Daher wurde in dieser Veranstaltung bereits mehrfach das entsprechende Werkzeug aSQLg eingesetzt und dieser Einsatz empirisch untersucht. Die Ergebnisse basieren auf Befragungen von Studierenden im Sommersemester 2012 und 2013. Zu dieser Zeit wurde das System nicht im Kontext eines vollwertigen LMS eingesetzt, sondern als eigenständiges Werkzeug angeboten. Insgesamt beteiligten sich 39 Studierende an der Umfrage im Sommersemester 2012, die im folgenden genauer erläutert wird; dies entspricht in etwa 80 % der Teilnehmenden. Die Ergebnisse (vgl. auch Abb. 4.3) zeigen, dass über 50 % sich mindestens

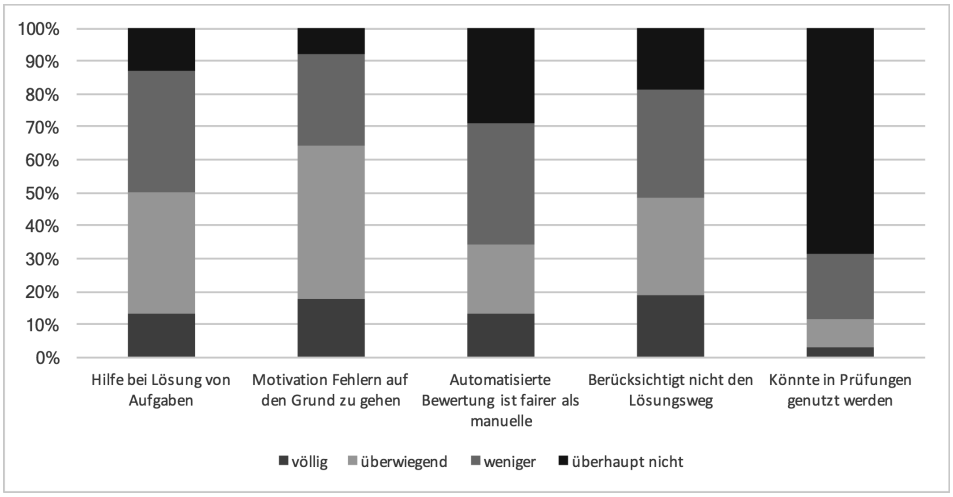

Abbildung 4.3: Einschätzung der Studierenden zur automatisierten SQL-Bewertung in der Veranstaltung Informationssysteme I des Studiengangs BIN (n=39)

überwiegend bei der Lösung der Aufgaben unterstützt fühlten. Dazu passt die Angabe, dass etwa 40 % das unmittelbare Feedback von aSQLg nach Einreichen einer Lösung als die wertvollste Funktion bewerteten.

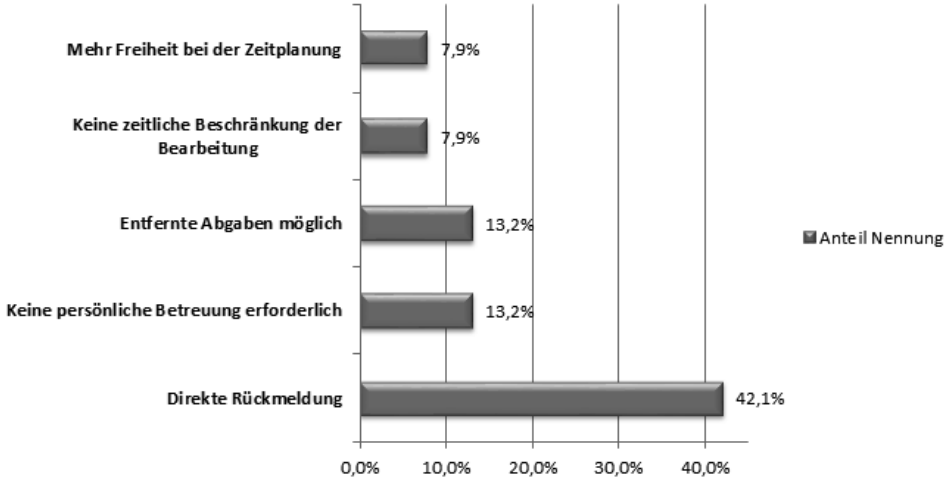

Abbildung 4.4: Zustimmung zu positiven Aspekten in der Veranstaltung Informationssysteme I des Studiengangs BIN (n=39)

Die Steigerung der Motivation bei der Bearbeitung der Aufgaben sowie die tiefer gehende Fehlersuche bei initialen Problemen ist insbesondere im Bereich der SQL-Ausbildung problematisch. Insofern ist es wichtig, dass etwa 70 % der Studierenden sich durch das Werkzeug motiviert fühlten, verbliebene Probleme in ihren Abgaben zu beheben. Möglicherweise hat damit das Werkzeug auch zur insgesamt verbesserten Bewertung der Lehrveranstaltung beigetragen.

Im Bereich der kritischen Anmerkungen ist insbesondere die Befürchtung zu nennen, dass manuelle Bewertungen fairer sind (70 %) und dass bei automatisierter Bewertung sehr ergebnisorientiert und weniger lösungswegorientiert (50 %) bewertet wird. Diese sind allerdings nur für Szenarien relevant, in denen die Bewertungen vom Lehrenden weiterverwendet werden. Für die reine Unterstützung der Studierenden bei der Lösungsfindung ist dies hingegen nicht von Bedeutung.

Die wichtigsten genannten Vorteile des Systems (vgl. Abb. 4.4) neben der unmittelbaren Rückmeldung auf die Lösung waren die Möglichkeit selbstbestimmter zu lernen (in Bezug auf Raum und Zeit) sowie die Möglichkeit einer entfernten Abgabe von Aufgaben. Insofern treten auch hier sowohl didaktische wie auch organisatorische Vorteile auf. Das Werkzeug selbst zeigte nur sehr geringe Fehlerraten und ca. 60 % würden es auch in Datenbankveranstaltungen zu fortgeschrittenen Themen einsetzen wollen. 40 % der Studierenden können sich einen Einsatz in anderen Informatikveranstaltungen vorstellen.

Bei den negativen Aspekten der automatisierten SQL-Bewertung wurden neben didaktischen (bspw. Nichtberücksichtigung der Lösungswege) vor allem organi-

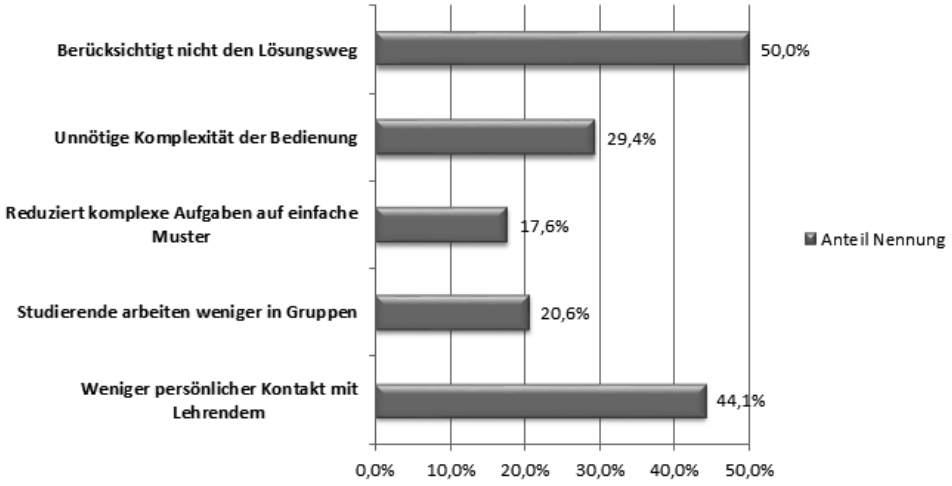

Abbildung 4.5: Zustimmung zu negativen Aspekten in der Veranstaltung Informationssysteme I des Studiengangs BIN (n=39)

satorische Themen (bspw. Befürchtungen zur zunehmenden sozialen Isolation der Lernenden) genannt (vgl. Abb. 4.5). Diese Aspekte hängen nicht unmittelbar mit dem Werkzeug selbst zusammen, sondern eher mit der Betreuungsform im Allgemeinen. Der Einsatz sollte also daher ergänzend zu Präsenzveranstaltungen denn ersetzend erfolgen. Dies wird auch durch die Zurückhaltung bei der Einschätzung der Eignung für Prüfungen deutlich.

4.5 Fazit und Ausblick

Als primäres didaktisches Einsatzszenario automatisierter Programmbewertung bei der Vermittlung der SQL ist das formative Assessment im Rahmen begleitender Übungen zur Vermittlung von Anwendungskompetenz zu nennen. Abhängig von der Feedbackqualität bietet sich der Einsatz von SQL-Gradern vor allem als ergänzendes Angebot zu betreuten Übungen an. Aber auch summatives Assessment (E-Klausuren), die Vorbereitung hierauf sowie Selbsteinschätzung der Studierenden sind sinnvoll umsetzbare Szenarien.

Wird automatisierte Bewertung von SQL-Anfragen in E-Klausuren eingesetzt, sollte bei der Aufgabenstellung sowie der Auswahl des Graders berücksichtigt werden, dass viele Grader auf eine Syntaxprüfung und einen Vergleich der Abfrageresultate fokussieren, und daher ggf. Lösungen akzeptiert werden, die zwar das gewünschte Resultat produzieren, von der Aufgabenstellung gegebenenfalls so aber nicht intendiert waren. Produziert ein Grader unter Umständen Fehlbewertungen, sollten Lösungen, welche die volle Punktzahl nicht erreicht haben, ggf. noch einmal manuell geprüft werden.

Wichtige Erfolgsfaktoren des Einsatzes von SQL-Gradern sind zudem eine gute technische Unterstützung bei Fehlern und Bedienungsproblemen, eine möglichst nahtlose Integration in die gewohnte Arbeitsumgebung bzw. das an der jeweiligen Hochschule verwendete Lernmanagementsystem, eine möglichst abteilungs- bzw. hochschulweit einheitliche Präsentation über mehrere Veranstaltungen bzw. Programmiersprachen hinweg sowie der Einsatz von automatisierter Bewertung in mehreren Veranstaltungen (z. B. Programmieren, Theoretische Informatik, Modellierung).

Um die Effektivität des Einsatzes von SQL-Gradern in den oben genannten didaktischen Szenarien zu steigern, ist weitere Forschungs- und Entwicklungsarbeit notwendig, vor allem auf den Gebieten Feedbackqualität und semantische Analyse der Lösungen, Aufgaben-Repositories und Austauschstandards für SQL-Aufgaben, unterstützter Sprachumfang und LMS-Integration. Studierende würden im Rahmen des formativen Assessments von einem Feedback profitieren,

das nicht nur etwaige syntaktische Fehler auflistet, sondern bei fehlerhaften Lösungsvorschlägen die Fehler erklärt und hieraus abgeleitete Verbesserungsvorschläge gibt. Hierfür ist eine semantische Analyse der Lösung im Vergleich zur Musterlösung erforderlich, die zudem erlauben würde, von einer „ganz oder gar nicht"-Bewertung eines Lösungsvorschlages wegzukommen. Lehrende würden von online verfügbaren Aufgaben-Repositories profitieren, die im Falle von SQL-Aufgaben nicht nur die Aufgabenstellung sowie die Musterlösung beinhalten, sondern auch die zugrundeliegende Beispieldatenbank, Bewertungsvorschläge und Feedbackvorgaben. Idealerweise lassen sich die Aufgaben des Repositories für beliebige SQL-Grader einsetzen, was wiederum ein standardisiertes Aufgabenaustauschformat erfordert (siehe hierzu Kapitel 24). Weiterhin wäre die vollständige Unterstützung auch von DDL, DML und DCL (Data Control Language) erforderlich, um das gesamte Spektrum von SQL-Statements abfragen zu können. Für Aufgaben im Rahmen des summativen Assessments wären parametrierbare, d. h. für jeden Studierenden individualisierte Aufgaben hilfreich, um Plagiate zu verhindern. Für Studierende und Lehrende gleichermaßen hilfreich wäre die nahtlose Integration der SQL-Grader in das gewohnte Lernmanagementsystem.

Literatur für dieses Kapitel

[Abe+08] Alberto Abelló u. a. „LEARN-SQL: Automatic Assessment of SQL Based on IMS QTI Specification". In: *ICALT*. IEEE, 2008, S. 592–593.

[AKB01] Lorin W. Anderson, David R. Krathwohl und Benjamin Samuel Bloom. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Allyn & Bacon, 2001.

[Bha+15] Amol Bhangdiya u. a. „The XDa-TA system for automated grading of SQL query assignments". In: *2015 IEEE 31st International Conference on Data Engineering*. Apr. 2015, S. 1468–1471. DOI: 10.1109/ICDE.2015.7113403.

[DD97] Chris J. Date und Hugh Darwen. *A Guide To Sql Standard*. Bd. 3. Addison-Wesley Reading, 1997.

[Ifl+14b] Marianus Ifland u. a. „ÜPS – Ein autorenfreundliches Trainingssystem für SQL-Anfragen". In: *DeLFI 2014 – Die 12. e-Learning Fachtagung Informatik*. Bd. 233. LNI. GI, 2014, S. 259–264. ISBN: 978-3-88579-627-5.

[Mit03] Antonija Mitrovic. „An Intelligent SQL Tutor on the Web“. In: *I. J. Artificial Intelligence in Education* 13.2-4 (2003), S. 173–197.

[PK09] Claus Pahl und Claire Kenny. „Interactive Correction and Recommendation for Computer Language Learning and Training“. In: *IEEE Transactions on Knowledge and Data Engineering* 21.6 (Juni 2009), S. 854–866. DOI: 10.1109/TKDE.2008.144.

[Sad+04] Shazia Sadiq u. a. „SQLator: an online SQL learning workbench“. In: *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*. ITiCSE ’04. ACM, 2004, S. 223–227. DOI: 10.1145/1007996.1008055.

[Sol+06] Josep Soler u. a. „A Web-based tool for teaching and learning SQL“. In: *International Conference on Information Technology Based Higher Education and Training, ITHET*. 2006.

[Stö+13] Andreas Stöcker u. a. „Evaluation automatisierter Programmbewertung bei der Vermittlung der Sprachen Java und SQL mit den Gradern *aSQLg* und *Graja* aus studentischer Perspektive.“ In: *DeLFI 2013 – Die 11. E-Learning Fachtagung Informatik*. Bd. 218. LNI. GI, 2013, S. 233–238.

[UGB13] Anja Ufert, Andreas Grillenberger und Torsten Brinda. „eledSQL – Entwicklung und Erprobung einer webbasierten Lernumgebung für Datenbanken und SQL“. In: *DeLFI 2013 – Die 11. E-Learning Fachtagung Informatik*. Bd. 218. LNI. GI, 2013, S. 167–178.