

Web-basiertes SQL-Training im Bildungsportal Sachsen*

Dieter Sosna, Erhard Rahm

Universität Leipzig, Institut für Informatik

<http://dbs.uni-leipzig.de>

Kurzfassung:

Der Betrag skizziert die Konzeption und Realisierung eines interaktiven, web-basierten SQL-Trainers. Der SQL-Trainer wurde im Rahmen des Verbundprojekts „Bildungsportal Sachsen“ entwickelt und ist unter der URL <http://sql-trainer.uni-leipzig.de> mit einem Gast-Login zum Testen erreichbar.

1. Einführung

Die Datenbankanfragesprache SQL hat sich zum Standardinstrument bei der Arbeit mit relationalen Datenbanken etabliert. Durch die weitreichende Standardisierung, v.a. im Rahmen von SQL-92 und SQL:1999, wurde eine sichere Basis gelegt, die von den meisten kommerziellen sowie freien Datenbankverwaltungssystemen (DBVS) als Anfrageschnittstelle unterstützt wird. Hinsichtlich der Umsetzung der Standards, insbesondere von SQL:1999, gibt es jedoch vielfach noch Verbesserungsbedarf.

Dessen ungeachtet ist es unumgänglich, den Umgang mit dieser Anfragesprache in den praktischen Teil der Informatikausbildung an Universitäten, Fachhochschulen und sonstigen Ausbildungsstätten aufzunehmen. Dabei zeigt sich in Analogie zum Erlernen einer Programmiersprache, daß der Lernerfolg sowohl von einer guten Vermittlung der Sprachkonzepte und der Sprachbeschreibung als auch, und dies vor allem, von der Möglichkeit des praktischen Übens abhängt. Diese Möglichkeit soll mit der hier vorgestellten web-basierten interaktiven Lernumgebung geboten werden.

Im folgenden beschreiben wir in knapper Form wesentliche Anforderungen, die bereitgestellte bzw. geplante Funktionalität sowie Realisierungsaspekte. Der SQL-Trainer befindet sich an der Universität Leipzig für Studierende der Informatik und Wirtschaftsinformatik bereits im praktischen Einsatz.

2. Anforderungen und Funktionalität

Wesentliche Zielsetzung für den SQL-Trainer war die Unterstützung einer einfach nutzbaren und flexiblen SQL-basierten Datenbankabfragemöglichkeit unter realitätsnahen Bedingungen. Dies implizierte einerseits die Forderung nach einer web-basierten Realisierung, bei der das System von jedem internet-fähigen Rechner über einen Standard-Browser genutzt werden kann. Insbesondere sollte keinerlei lokale Zusatz-Software auf dem Nutzerrechner installiert oder sogar eine eigene Datenbank eingerichtet werden müssen, wie bei anderen Lernsystemen oft erforderlich. Hohe Flexibilität bedeutet, dass nicht nur vordefinierte bzw. gering parametrisierbare Anfragen zugelassen werden sollen, sondern dass ein Großteil des SQL-Sprachumfangs interaktiv und unbeschränkt erprobt werden kann. Wesentlich für eine hohe Realitätsnähe ist die Ausführung der SQL-Anfragen auf einer realen Datenbank nicht-trivialer (aber auch nicht un-

*. Die Realisierung erfolgte im Projekt „Internetgestützter SQL-Trainer“ des vom Sächsischen Ministeriums für Wissenschaft und Kunst (SMWK) geförderten Verbundprojekts „Bildungsportal Sachsen“.

nötig hoher) Schemakomplexität und Datenmenge unter Verwendung eines realen Datenbanksystems. Kriterien bei der Auswahl eines geeigneten SQL-Servers sind Sprachumfang, Standard-Konformität, verfügbare Anbindung an Programmiersprachen und Web-Server und nicht zuletzt der Preis.

Diese Forderungen werden bereits mit der Basisfunktionalität des Systems, dem sogenannten **freien Üben** auf einer Übungsdatenbank unterstützt. Der Nutzer kann dabei nahezu unbeschränkt (lesende) Anfragen formulieren, zur Ausführung bringen sowie die Ergebnisse (bzw. Fehlernachrichten) angezeigt bekommen. Zur Unterstützung der Anfragemöglichkeiten kann das Schema der Datenbank jederzeit angezeigt werden. Das System ermöglicht ferner die Unterscheidung verschiedener Benutzer und Nutzergruppen, denen mehrere Datenbanken zur Auswahl und Bearbeitung angeboten werden können. Für den Gastzugang ist lediglich eine Übungsdatenbank freigeschaltet.

Als zweite Nutzungsstufe bietet der SQL-Trainer für das Selbststudium sowie als Nachschlagemöglichkeit ein **aktives SQL-Tutorium** an. „Aktiv“ bedeutet hierbei, daß die meisten Beispiele zu den erklärten SQL-Konstrukten auf der Übungsdatenbank per Knopfdruck ausgeführt werden können. Der Nutzer kann die Beispielanfragen auch modifizieren, um die Auswirkungen der Modifikationen auf das Ergebnis unmittelbar prüfen zu können und so zu einem vertieften Verständnis der Funktionsweise der Konstrukte zu kommen.

Derzeit geplant, jedoch noch nicht realisiert, ist die Unterstützung eines universitären Übungsbetriebs mit Übungsblättern sowie teil-automatischer Korrekturmöglichkeit von Lösungen (s. Kap. 4).

3. Realisierung

Als DB-Server (Backend) dient ein relationales DBVS. Hier besteht eine relativ große Auswahl, die zu Gunsten von Postgres entschieden wurde. Die Verwendung eines anderen Systems ist relativ einfach umsetzbar.

Den Forderungen nach minimalem Aufwand und der Nutzung von Standardsoftware beim Anwender führten zu einer web-basierten Client-Server-Lösung, die sich aus Sicht des Nutzers als Web-Server darstellt, auf den er mit seinem Web-Browser zugreift. Zur Wiedererkennung eines Nutzers sowie zur Sitzungskontrolle werden derzeit Cookies eingesetzt. Mit dem sicheren HTTP (HTTPS) wird der Schutz der Übertragung zwischen Client und Server erreicht.

Die Implementierung der die SQL-Trainer-Funktionalität realisierenden Middleware erfolgte mit der Programmiersprache Java, die einen problemlosen DBS-Zugang ermöglicht (JDBC). Als HTTP-Server wird Apache eingesetzt. Die bei der interaktiven Nutzung des SQL-Trainers verwendeten dynamischen Web-Seiten werden durch Java-Server-Pages erzeugt. Dem Web-Server wird ein Servlet-Container beigelegt, in dem die Erzeugung der Seiten aus Bausteinen erfolgt. Im Sinne einer strikten Modularisierung übernehmen die dort lokalisierten Programmkomponenten nur die Erzeugung der dynamisch generierten Seiten, d.h. sie übernehmen die Ausgabegestaltung für an anderer Stelle im System gewonnene Information. Aus seiner Sicht ist der Servlet-Container Client einer weiteren Komponente, des sogenannten Trainer-Servers.

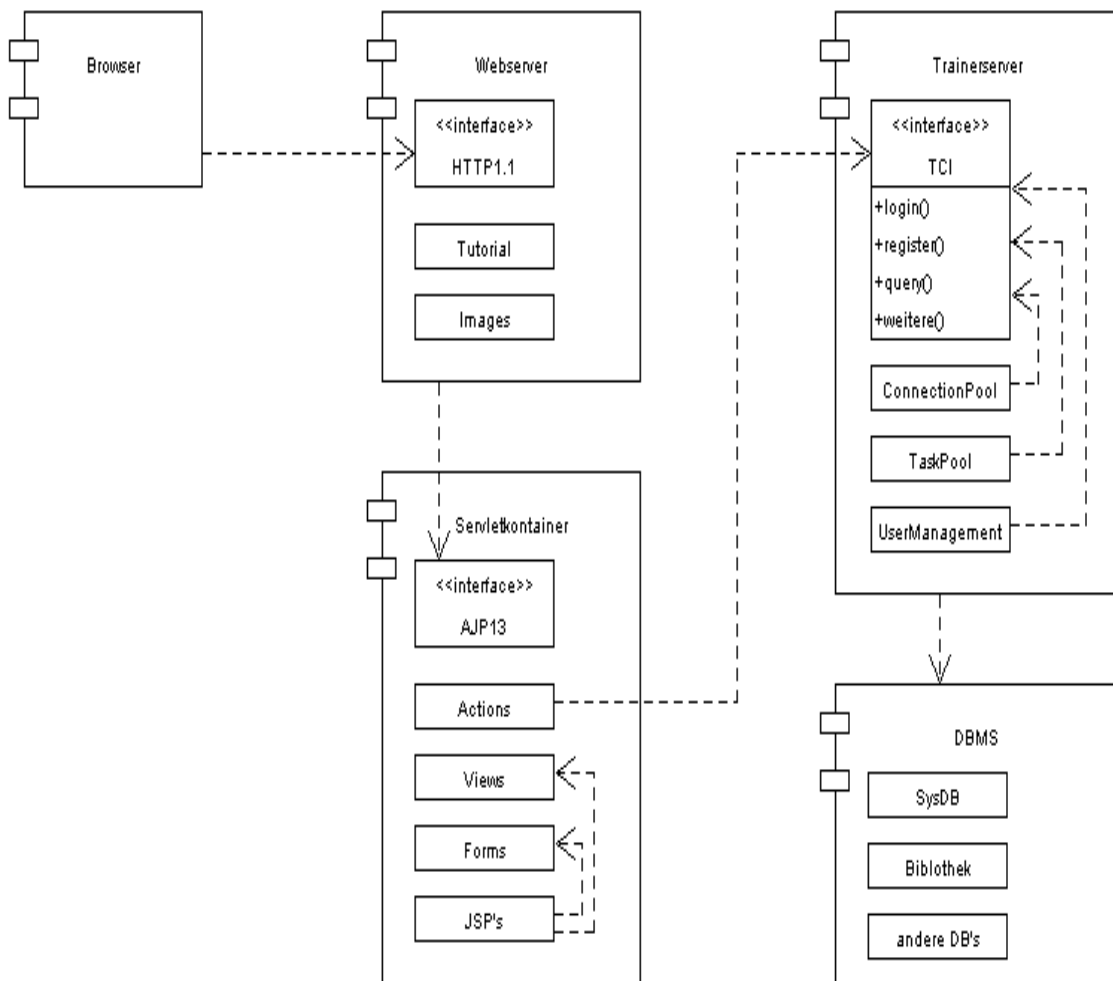


Abbildung 1: Struktur des SQL-Trainers

Der Trainer-Server ist eine Java-Applikation, die Aufträge aus dem Servlet-Container entgegennimmt und die Ergebnisse der Auftragsbearbeitung zurückliefert. Dabei nimmt er die Umsetzung des zustandslosen HTTP in zustandsbasierte Datenbankverbindungen vor. Zu diesem Zweck realisiert er eine Nutzer- und Sessionverwaltung, in der Informationen über die Nutzer, den Stand ihrer Aktivitäten und ihre Arbeitsergebnisse festgehalten werden. Sie stützt sich ihrerseits auf eine Datenbank ab. Der Zugriff aus dem Servlet-Container auf den Trainer-Server erfolgt über das RMI, so daß eine verteilte Implementierung möglich ist, wenn dies etwa durch eine starke Last geraten erscheint. Außerdem unterstützt die strenge Strukturierung die verteilte Arbeit des Entwicklerteams.

Die Bereitstellung der Inhalte des Tutorials erfolgt in Form einer XML-Datei außerhalb des Trainers. Während eines Ladevorgangs werden daraus statische HTML-Seiten generiert. Bei diesem Generierungsprozeß werden auch aus den im XML-Text enthaltenen SQL-Anfragebeispielen die interaktiv ausführbaren SQL-Statements des Tutorials erzeugt.

4. Geplante Weiterentwicklungen

Für den vorlesungsbegleitenden Übungsbetrieb ist die termingerechte Bearbeitungen von Aufgabenblättern typisch. Dies soll in Zukunft vom SQL-Trainer unterstützt werden. Dazu werden

Einzelaufgaben zu Aufgabenblättern zusammengefaßt. Diese werden für einen vom Lektor bestimmten Zeitraum zur Bearbeitung freigegeben. Während dieser Zeit können die Aufgaben beliebig bearbeitet werden, wobei der jeweils letzte Stand gespeichert bleibt. Es werden also beliebige Korrekturen der Lösung durch den Nutzer akzeptiert. Mit Ablauf der Bearbeitungsfrist wird der Stand eingefroren und bewertet (syntaktische Korrektheit, Vergleich der Anfrageergebnisse mit einem Lösungsmuster).

Bei Analyse dieser Zielstellung zeigte es sich, daß hier Erweiterungen auf andere Aufgabentypen möglich sind, sofern sich ein Algorithmus zur automatischen Bewertung angeben läßt. Zu dieser Aufgabenklasse gehören Multiple-Choice-Aufgaben mit ein- und mehrwertiger korrekter Antwort und bestimmte Typen von Einsetzaufgaben, bei denen auf Grund der Aufgabenstellung und eines als bekannt vorausgesetzten Fachwortschatzes eine eindeutige Wortwahl erwartet werden kann. Die Korrektheit der Antworten kann hier vom System durch Vergleich mit Antwortmustern bewertet werden, die Umsetzung in eine Leistungsbewertung erfolgt mittels einer vom Lektor vorgegebenen Gewichtsfunktion.

Die Realisierung dieser Überlegungen soll in der folgenden Phase des Verbundprojekts „Bildungsportal Sachsen“ erfolgen.