

Recherche von interaktiven Lehr- und Lernmaterialien für das Fach Datenbanksysteme von Timo Ribbers und Joshua Westerheide

Arbeitspapier des Lehrgebiets
Datenbanken und E-Business
No. 1/2023

herausgegeben von
Thomas C. Rakow

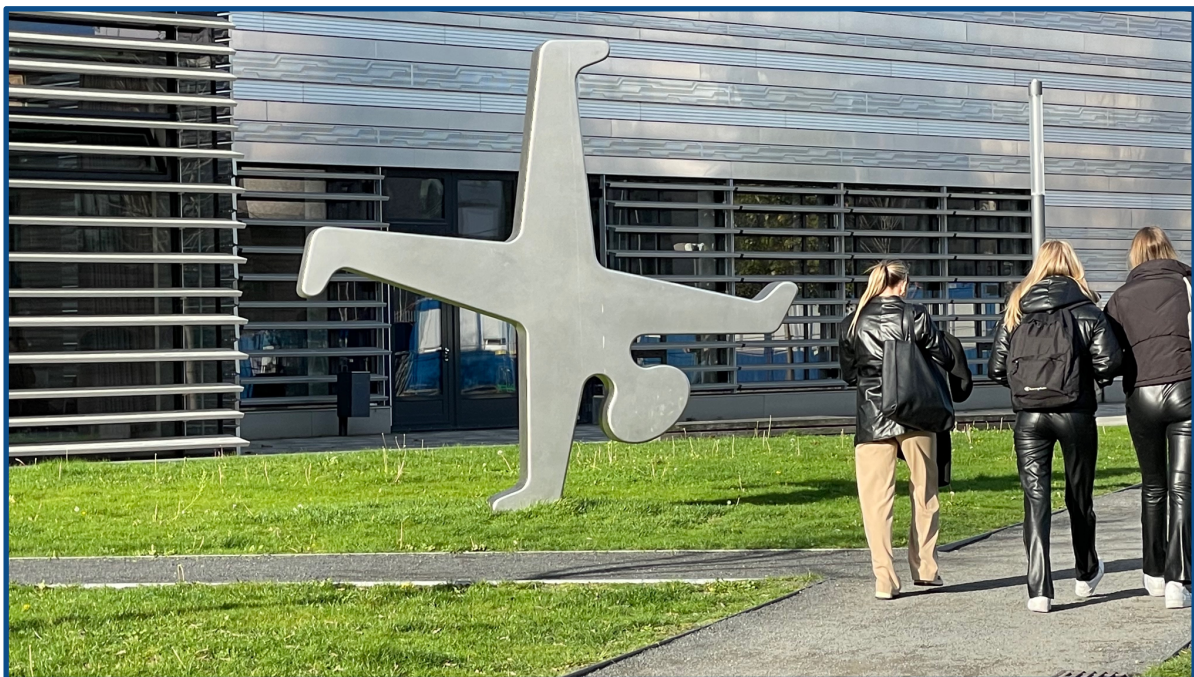


Bild: Radschläger auf Campus HSD.
© Thomas Rakow, 2023.

© Die Autoren 2023

Open Access: Dieses Werk ist lizenziert unter CC BY-SA 4.0. Um eine Kopie dieser Lizenz anzusehen, besuchen Sie <https://creativecommons.org/licenses/by-sa/4.0/>. Die Bedingungen der Creative-Commons-Lizenz gelten nur für Originalmaterial. Ausgenommen von der CC BY-SA 4.0 Lizenz sind die verwendeten Logos sowie alle anders lizenzierten Inhalte. Die Wiederverwendung von Material aus anderen Quellen (gekennzeichnet mit Quellenangabe) wie z. B. Schaubilder, Abbildungen, Fotos und Textauszüge erfordert ggf. weitere Nutzungsgenehmigungen durch den jeweiligen Rechteinhaber*innen.

Zum Geleit

Der Einsatz digitaler Lehr- und Lernmaterialien in der Lehre bietet vielfältige Möglichkeiten für neue Szenarien: Verfügbarkeit jederzeit und überall, Anpassbarkeit, insbesondere bei der individuellen Rezeptionsgeschwindigkeit, automatisierbare Überprüfung von Lösungen, Nachvollziehbarkeit der Nutzung sowie einfache Reproduzierbarkeit.¹ Die Materialien können aus Präsentationen, Skripten und Aufgaben, Umfragen und Tests bestehen, die in Lernmanagementsystemen den Lernenden angeboten werden. Die Formen der Lizenzierung, Nachhaltigkeit und Zugänglichkeit sowie die Möglichkeiten der Kontextualisierung und des Inhalts, die technische Anpassungsfähigkeit sowie die Einhaltung des Datenschutzes auf Plattformen sind die bestimmenden Faktoren für den Austausch digitaler Inhalte in der Lehre.²

In dieser Arbeit werden wissenschaftliche Artikel als auch Tools zur Lehre im Fach Datenbanken recherchiert, beschrieben und kategorisiert. Herr Ribbers und Herr Westerheide haben in nachvollziehbarer Weise die teilweise schwer und unterschiedlich beschriebenen Tools in eine übersichtliche Darstellung unter den oben genannten Anforderungen gebracht, insbesondere der freien Verfügbarkeit und Anpassbarkeit. Soweit vorhanden werden in den Artikeln berichtete Erfahrungen in der Verwendung der Tools dargestellt.

Herr Ribbers und Herr Westerheide haben diese Studienarbeit als Prüfungsleistung im B.Sc. Medieninformatik der Hochschule Düsseldorf erstellt.

Düsseldorf, 12.07.2023

Professor Dr.-Ing. Thomas C. Rakow

¹ Focken, M.: OER-Nutzung durch Lehrende. In: Thomas Rakow (Hrsg.): Literaturrecherche zum Einsatz offener digitaler Lehr- und Lernmaterialien, Düsseldorf, Dezember 2021. <https://doi.org/10.20385/opus4-3372>.

² Rakow, T.C.; Faeskorn-Woyke, H.; Saatz, I.M.; Knolle, H.: Es EILD – Anforderungen an die Publikation freier Lerneinheiten (OER) im Fach Datenbanken. *Datenbank-Spektrum*, 2(12) 111-120. Springer-Verlag, Berlin 2021. <https://doi.org/10.1007/s13222-021-00373-z>

Zusammenfassung

Recherche von interaktiven Lehr- und Lernmaterialien für das Fach
Datenbanksysteme

Timo Ribbers, Joshua Westerheide

Die Lehre von Datenbanksystemen kann durch interaktive Lehr- und Lernmaterialien ergänzt werden, um Studierende besser an die Themenbereiche heranzuführen. Diese Arbeit dokumentiert die systematisch durchgeführte Recherche interaktiver Materialien und stellt die Rechercheergebnisse vor. Dabei wurden sowohl einzelne Anwendungen als auch wissenschaftliche Veröffentlichungen berücksichtigt. Von den insgesamt 59 gefundenen Materialien wurde eine Handvoll zur näheren Betrachtung ausgewählt. Im Detail wird auf deren Zielsetzung, Umsetzung und die Bedienung eingegangen. Die Materialien befassen sich mit den Themenbereichen SQL, ERM/UML, Normalisierung, Relationale Algebra und Weiteren. Auch bei der Umsetzung sind Unterschiede hinsichtlich der Herangehensweise und der ausgewählten Lernmethoden ersichtlich. Im Rahmen wissenschaftlicher Veröffentlichungen wurden Experimente mit Studierenden durchgeführt und deren Lernfortschritt ausgewertet. Aus dem Feedback ist ein grundsätzliches Interesse an interaktiven Lernmaterialien zu erkennen. Gleichzeitig konnten Studierende mithilfe dieser Materialien in Tests bessere Ergebnisse erzielen.

Schlüsselbegriffe: DSE, DBMS, Lehre, Datenbanken, Datenbanksysteme, OER, interaktiv

Anzahl Wörter: 11 361

Abstract

Research of interactive learning materials for database systems education

Timo Ribbers, Joshua Westerheide

In database systems education, interactive learning materials may be used to help student better understand topics. This paper documents the systematically conducted research of interactive materials and presents the research results. Both individual applications and scientific publications were considered. Of the 59 materials found, a handful were selected for closer examination. Their objectives, implementation, and usability are discussed in detail. Topics covered are SQL, ERM/UML, database normalization, relational algebra and many more. Furthermore there are differences in the approach and selected learning methods. Experiments conducted with students yielded positive feedback: Not only did students indicate a general interest in interactive learning materials, their grades also improved in exams thanks to these materials.

Keywords: DSE, DBMS, education, database, database systems, OER, interactive
Word count: 11 361

Inhaltsverzeichnis

1	Einleitung	1
2	Recherche	4
2.1	Verwandte Arbeiten	4
2.2	Wissenschaftliche Veröffentlichungen	5
2.2.1	Rechercheplattformen	6
2.2.2	Suchparameter	7
2.2.3	Ergebnis	7
2.2.4	Auswahlgründe	9
2.3	Interaktive Lehr- und Lernmaterialien	11
2.3.1	Portale	11
2.3.2	Ergebnis	12
2.3.3	Auswertung	13
2.4	Sonstige	20
2.4.1	Onlinekurse	20
3	Ergebnisse	21
3.1	Wissenschaftliche Arbeiten	21
3.2	Materialien	23
3.2.1	Entity-Relationship-Modell	23
3.2.2	UML-Klassendiagramme	29
3.2.3	Normalisierung	33
3.2.4	Datenstrukturen	35
3.2.5	Structured Query Language	38
3.2.6	Relationale Algebra	46
3.2.7	Themenübergreifende Materialien	48
4	Fazit	50
5	Ausblick	54

Abbildungsverzeichnis

2.1	Vorgehen mit rekursiver Recherche	6
2.2	Übersicht über die Methodiken	13
2.3	Anzahl an Methodiken pro Material	13
2.4	Übersicht über die Themenbereiche	14
2.5	Anzahl an Themen pro Material	14
2.6	Lizenzen in den Materialien	14
3.1	Screenshot zeigt die Bearbeitung einer Aufgabe in KERMIT [aus MS16, Abbildung 1.]	24
3.2	MonstER Park Benutzeroberfläche	26
3.3	Ansicht einer Aufgabe im ER-Trainer	28
3.4	Überprüfung der Eingabe im ER-REL-Trainer	29
3.5	AutoER Webinterface: Bearbeitung einer Aufgabe [aus FUL22, S. 628]	31
3.6	Interface zum Erstellen eines Unified Modeling Language (UML)-Klassendiagramms mit Aufgabenstellung in ACME-DB [aus Sol+10, Abbildung 6.]	32
3.7	Verschiedene Interfaces des Normalization Educational Tool	35
3.8	Bearbeitung einer Aufgabe im NF-Trainer	36
3.9	Darstellung eines Binärbaums in VisuAlgo	37
3.10	SQL-Tutor – Webinterface nach der Integration von Gamification Elementen [TMS20]	39
3.11	SQL-Trainer EILD.nrw – Data Query Language (DQL)-Aufgabe	41
3.12	SQLBolt – Erklärungs- und Übungsteil der „SQL Lesson 1“	43
3.13	Ein DB-Kreuzworträtsel zu den SQL-Grundlagen	44
3.14	freeCodeCamp – „Learn SQL by Building a Student Database: Part 1“ eine SQL-Übung	45
3.15	Relax – Beschreibung und Ergebnis eines Ausdrucks aus der Movie Datenbank	47
3.16	IRA – Eine Relation zwischen zwei Mengen in RA	48
3.17	MCT-Trainer – Eine Multiple Choice Frage aus dem Kapitel „Einführung in die Grundbegriffe der Datenbanken“	49

Tabellenverzeichnis

2.1	Suchbegriffe für die Recherche	8
2.2	Ergebnis der Literaturrecherche	10
2.3	Ergebnis der Materialrecherche	16

Abkürzungsverzeichnis

DBMS	Database Management Systems
RDBMS	Relational Database Management Systems
OER	Open Educational Resources
DSE	Database System Education
OER	Open Educational Resources
LMS	Lernmanagementsystem
ERM	Entity-Relationship-Modell
ER-Diagramm	Entity Relationship Diagramm
ACM	„Association for Computing Machinery“
IEEE	„Institute of Electrical and Electronics Engineers“
SQL	Structured Query Language
UML	Unified Modeling Language
ORCA	Open Resources Campus NRW
CASE	Computer-aided software engineering
IDE	Integrated Development Environment
OOP	Objectorientierte Programmierung
DSL	Domain-specific Language
ITiCSE	Innovation and Technology in Computer Science Education
SIGCSE	Special Interest Group Computer Science Education
ICTG	Intelligent Computer Tutoring Group
ISO	„International Organization for Standardization“
DDL	Data Definition Language
DML	Data Manipulation Language
DQL	Data Query Language
DCL	Data Control Language
CBM	Constraint-Based Modeling
EERM	Erweitertes Entity-Relationship-Modell
BCNF	Boyce-Codd Normalform
MOOCs	Massive Open Online Courses

Kapitel 1

Einleitung

Datenbanken sind ein essenzieller Bestandteil verschiedener Informationssysteme, um Daten strukturiert speichern und verarbeiten zu können. Datenbankmanagementsysteme (engl. Database Management Systems (DBMS)) übernehmen hierbei die eigentliche Arbeit der Verwaltung jener Daten und bieten Entwicklerinnen und Entwicklern eine einheitliche Schnittstelle, um mit diesen Daten interagieren zu können. Dennoch ist es wichtig die Grundlagen von Datenbanken und DBMS zu verstehen und Methoden zu erlernen, um diese erfolgreich integrieren zu können.

Ishaq et, al. [Ish+22] fassen die Lehre von Datenbanksystemen unter dem Begriff *Database System Education (DSE)* zusammen. Sie verstehen darunter die Grundlagen und Implementation von Datenbanksystemen, als auch Konzepte zur Datenmodellierung. Das Fach Datenbanksysteme ist Bestandteil vieler Informatik-Studiengänge [vgl. Ish+22, S. 2681]. Üblicherweise werden dafür konventionelle Lehrmethoden, wie der Frontalunterricht, angewendet. In praktischen Übungen wird das theoretische Wissen im Anschluss oftmals vertieft.

Interaktive Lehr- und Lernmaterialien können dabei komplementär eingesetzt werden, um abstrakte und komplexe Themen verständlicher zu erklären. Durch integrierte Quiz kann der Wissensstand und Lernfortschritt von Studierenden abgefragt werden. Mit interaktiven Animationen können Algorithmen und Mechanismen visualisiert werden. Gleichzeitig haben interaktive Materialien die Möglichkeit, individuell auf Studierende angepasst zu werden. Anhand des Vorwissens kann bspw. der Schwierigkeitsgrad der Aufgaben angehoben werden. Eine leichte Zugänglichkeit (z. B. online) und zeitunabhängige Bearbeitung, ermöglichen es Studierenden ihr eigenes Lerntempo zu bestimmen. Durch eine automatisierte Auswertung von Antworten kann unmittelbar Feedback gegeben werden.

Nicht nur Studierende können von interaktiven Lehr- und Lernmaterialien profitieren, sondern auch Lehrende. So können Ergebnisse aus Quiz und Übungen schneller ausgewertet werden. Damit kann ein Überblick über den gesamten Lernfortschritt aller Studierenden gewonnen werden. Einzelne Inhalte können jederzeit erweitert oder angepasst werden. Auch lassen sich interaktive Lehrmaterialien in bestehende Lernmanagementsysteme (LMS) integrieren.

Da diese Materialien teilweise öffentlich zugänglich sind und unter einer freien Lizenz stehen, spricht man auch von Open Educational Resources (OER). Genauer definiert die UNESCO¹ darunter Materialien jeglicher Art, deren freie Nutzung, Anpassung und Verbreitung durch Dritte ohne monetäre Kosten, gestattet ist².

Experimente zeigen, dass Studierende mithilfe von interaktiven Lernmaterialien Sachverhalte einfacher und schneller verstehen. Atcharyachanvanich et, al. [ANJ19] untersuchten die Noten von 152 Studierenden bei mehreren Tests zu SQL-Abfragen. Sie haben zuvor eigene interaktive Lernmaterialien erstellt. Dabei stellten sie fest, dass Studierende, die diese Materialien zum Lernen verwenden durften, nicht nur besser abschnitten, sondern auch sich selbst verbesserten [vgl. ANJ19, S. 55001 f.]. Bei einer Umfrage von Kleerekoper und Schofield [KS18] unter Studierenden, gaben diese an, durch interaktive Lernmaterialien, motivierter zu sein und mehr Zeit investieren zu wollen [vgl. KS18, S. 89 ff.]. In einer weiteren Datenerhebung über eine Laufzeit von vier Jahren zeigen Soler et, al. [Sol+09], dass Studierende Aufgaben zu den Themen ER-Diagramm, relationales Datenbankschema und Normalisierung mithilfe einer Web-basierten E-Learning Anwendung in Tests bessere Ergebnisse erzielen konnten [siehe Sol+09, S. 38 f.].

Ziel dieser Arbeit ist es interaktive Lehr- und Lernmaterialien zum Fach Datenbanksysteme zu recherchieren und zu beschreiben. Dadurch soll ein Überblick über die aktuellen Anwendungen und den Fortschritt in der Lehre geschaffen werden. Die Recherche wird systematisch durchgeführt und dokumentiert. Ebenfalls werden wissenschaftliche Veröffentlichungen hinzugezogen.

In Kapitel 2 wird zunächst das Vorgehen bei der Recherche und die Auswahl der Materialien beschrieben. Es wird zwischen wissenschaftlichen Veröffentlichungen und Materialien unterschieden. Die Materialien werden anhand verschiedener Merkmale kategorisiert und die Rechercheergebnisse eingeordnet. Anschließend werden die einzelnen Materialien und wissenschaftliche Veröffentlichungen in Kapitel 3 vorgestellt. Für eine bessere Übersicht, ist Abschnitt 3.2 in unterschiedliche Themenbereiche unterteilt. Zum Schluss wird das Ergebnis der Recherche in Kapitel 4

¹United Nations Educational, Scientific and Cultural Organization

²siehe Definition in UNESCO. *Open Educational Resources*. Deutsch. 2023. URL: <https://www.unesco.de/bildung/open-educational-resources>.

Kapitel 1. Einleitung

zusammen gefasst und in Kapitel 5 ein Ausblick auf zukünftige Entwicklungen und mögliche Verbesserungen gegeben.

Kapitel 2

Recherche

In diesem Kapitel wird die Vorgehensweise und Methodik bei der Literatur- und Materialrecherche vorgestellt. Der gesamte Prozess von initialer Recherche bis hin zur Auswahl der Literatur wird, orientiert am Kodex zur „Sicherung guter wissenschaftlicher Praxis“¹ der Deutsche Forschungsgemeinschaft e.V. , durchgeführt und dokumentiert. Dabei wird genauer auf die verwendeten Rechercheplattformen eingegangen und die jeweiligen Suchparameter aufgelistet. Ebenso werden die Kriterien, nach denen die entsprechende Literatur und Materialien ausgewählt wurde, erläutert. Die Recherche ist zwischen wissenschaftlichen Veröffentlichungen (Abschnitt 2.2) und Materialien (Abschnitt 2.3) aufgeteilt.

Voraussetzung für die Auswahl der Veröffentlichungen und Materialien ist eine einheitliche Definition interaktiver Lehr- und Lernmaterialien. Diese werden entweder eingesetzt, um Wissen zu vermitteln oder dieses zu vertiefen. Weiter müssen sie durch Benutzereingaben steuerbar sein.

2.1 Verwandte Arbeiten

Das Thema der Datenbanklehre (DSE) wird schon länger in der Wissenschaft behandelt. Eine der ersten Arbeiten wurde von Hall und Gordon [HG98] im Jahr 1998 veröffentlicht. Sie präsentieren ihre Ergebnisse zur Anwendung eines von ihnen entwickelten Systems zur Lehre des Entity-Relationship-Modell (ERM). Auf Konferenzen, in Symposien und in Fachzeitschriften werden in den darauffolgenden Jahren immer wieder neue Lehrmethoden und Materialien vorgestellt. Dazu

¹https://www.dfg.de/download/pdf/foerderung/rechtliche_rahmenbedingungen/gute_wissenschaftliche_praxis/kodex_gwp.pdf, aufgerufen am 27.03.2023

zählen die Innovation and Technology in Computer Science Education (ITiCSE)² Konferenz, das Special Interest Group Computer Science Education (SIGCSE)³ Forum und im deutschsprachigen Raum das Datenbank-Spektrum⁴ Journal der Fachgruppe Datenbanken und Information Retrieval der Gesellschaft für Informatik (GI) e.V. Die Intelligent Computer Tutoring Group (ICTG)⁵ an der University of Canterbury veröffentlicht seit 1998 unter Leitung von Tanja Mitrovic und Brent Martin Forschungsergebnisse zum Einsatz verschiedener Lernmaterialien in der DSE. Im Rahmen des Forschungsprojektes EILD.NRW⁶ werden seit 2020 von der Hochschule Düsseldorf, Fachhochschule Dortmund, Hochschule Bonn-Rhein-Sieg und der Technischen Hochschule Köln freie Lern- und Lehrmaterialien (OER) für das Fach Datenbanksysteme entwickelt.

Ishaq et, al. diskutieren die Forschung zu DSE in [Ish+22]. Sie untersuchen die Entwicklung und Anwendung verschiedener Materialien, Methoden und Lehrpläne. Außerdem gehen sie auf die Fortschritte in der DSE ein. Rakow et, al. [Rak+09] teilen Erfahrungen von der Anwendung von Werkzeugen in der Lehre. In [Shi22] und [Shi20a] untersucht Shin Lernmaterialien zu ERM und Structured Query Language (SQL) unter Anwendung von Lerntheorien.

2.2 Wissenschaftliche Veröffentlichungen

Das primäre Ziel dieser Recherche ist interaktive Lehr- und Lernmaterialien für das Fach Datenbanksysteme zu recherchieren. Dabei soll ein möglichst breites Spektrum der Themengebiete erfasst werden. Weil in wissenschaftlichen Arbeiten häufig jene Materialien ausgewertet werden oder deren Entwicklung beschrieben wird, fokussiert sich dieser Abschnitt zunächst auf wissenschaftliche Veröffentlichungen.

Durch die Verwendung verschiedener Rechercheplattformen (siehe Unterabschnitt 2.2.1) und unterschiedlicher Suchparameter (siehe Unterabschnitt 2.2.2) soll sichergestellt werden, dass das Themengebiet hinreichend erfasst werden konnte.

Um eine ausführliche Recherche zu betreiben, wurden als Erstes Oberbegriffe gesammelt, die das Thema beschreiben. Über Rechercheplattformen wurde damit initiale Literatur gefunden. Aus den Quellen wurden neue Begriffe deduziert, die fortan mit in der Suche einbezogen wurden. Zudem wurden die Quellen bereits erschlossener Veröffentlichungen berücksichtigt. Als weiteren Ansatz wurden wiederkehrende

²Homepage der ITiCSE: <https://iticse.acm.org/>

³Homepage der SIGCSE <https://sigcse.org/>

⁴Übersichtsseite des Datenbank-Spektrum: <https://www.springer.com/journal/13222>

⁵Projektseite der ICTG: <https://www.csse.canterbury.ac.nz/tanja.mitrovic/ictg.html>

⁶Projektseite: <https://eild.nrw>

Autoren, Institutionen und Konferenzen identifiziert, welche in der Suche separat berücksichtigt wurden.



Abbildung 2.1: Vorgehen mit rekursiver Recherche

So ergab sich ein rekursives Vorgehen bei der Recherche, welches so lange angewendet wurde, bis keine neuen Quellen mehr gefunden werden konnten oder der zuvor veranschlagte zeitliche Aufwand ausgeschöpft wurde. Abbildung 2.1 skizziert dieses Vorgehen grafisch.

Daraufhin wurden die bis zu diesem Zeitpunkt gefundenen Quellen auf ihre Relevanz untersucht. Die Relevanz der einzelnen Quellen wurde anhand des Titels, des Abstracts und den Ergebnissen der Evaluation beurteilt. Erst dadurch wurden sie für diese Arbeit in Erwägung gezogen.

2.2.1 Rechercheplattformen

Die Recherche wissenschaftlicher Veröffentlichungen wurde mithilfe der spezialisierten Suchmaschinen Google Scholar⁷ gestartet. Weiter wurden die Online-Portale verschiedener Organisationen und Verlage verwendet. Die digitale Bibliothek der „Association for Computing Machinery“ (ACM)⁸ bietet Zugang zu Veröffentlichungen im Bereich der Informatik. IEEE Xplore⁹ ist die digitale Bibliothek des

⁷<https://scholar.google.com>

⁸<https://dl.acm.org/>

⁹<https://ieeexplore.ieee.org/>

„Institute of Electrical and Electronics Engineers“ (IEEE). Beide Bibliotheken umfassen Fachzeitschriften, Konferenzen, Büchern und weitere Veröffentlichungen. Weiter beinhaltet IEEE Xplore auch Standards, die allerdings für die Recherche vernachlässigbar sind. Die Datenbank „dblp computer science bibliography“¹⁰ sammelt Informationen zu Veröffentlichungen in Fachzeitschriften und Symposien aus der Informatik, primär im deutschsprachigen Raum. Letztlich wurde Springer Link¹¹, die online Bibliothek des Springer Verlags, zur Recherche verwendet.

2.2.2 Suchparameter

Die Oberbegriffe, die zu Beginn gewählt wurden, ergeben sich zum einen aus dem Themengebiet Datenbanksysteme und zum Anderen aus den verschiedenen Arten der möglichen Umsetzungen (z. B. „Lernmaterial“, „Visualisierung“, „Werkzeug“, etc.). Zusammen mit weiteren Attributen, wie „E-Learning“, „online“ und „virtuell“ wurden die Sucheingaben geformt. Zusätzlich zu den deutschen Begriffen wurden auch deren englische Übersetzungen verwendet. Abkürzungen wurden ihren ausgeschriebenen Formen (bspw. SQL) vorgezogen, um die Suche zu vereinfachen.

Von dem Begriff und Thema „Datenbanksysteme“ lassen sich weitere Begriffe ableiten, z. B. „Datenbank“ und „Database Management Systems“. Um die Suche einzugrenzen, wurden teilweise die Unterthemen (z. B. SQL, UML, ER-Diagramm, usw.) explizit zur Sucheingabe hinzugefügt.

Tabelle 2.1 listet alle verwendeten Suchbegriffe und deren Übersetzungen auf. Sofern alternative Begriffe verwendet wurden, werden diese in der Spalte „Synonym“ aufgelistet.

Sucheingaben, die zu viele Ergebnisse lieferten, wurden konkretisiert oder durch andere Begriffe spezialisiert. Die Oberbegriffe sind in einer Vielzahl, für diese Recherche aber nicht relevanten, wissenschaftlichen Arbeiten wieder zu finden. Daher resultierten die initialen Suchbegriffe in sehr vielen Ergebnissen.

2.2.3 Ergebnis

Die relevanten Rechercheergebnisse werden in Tabelle 2.2 mit entsprechenden Eckdaten aufgelistet. In der Spalte „Auswahlgründe“ werden die Auswahlgründe kurz aufgelistet. In Unterabschnitt 2.2.4 werden diese näher erläutert. Es wurden nur deutsche und englische Veröffentlichungen in Erwägung gezogen. Die Anzahl der gefundenen englischsprachigen Quellen übertrifft jedoch die der deutschsprachigen.

¹⁰<https://dblp.uni-trier.de/>

¹¹<https://link.springer.com/>

Kapitel 2. Recherche
 2.2. Wissenschaftliche Veröffentlichungen

Schlagwort	Synonym	Übersetzung
Animation		animated
Datenbank	DB, (R)DBMS	database
Datenbanksysteme		database systems
E-Learning		e-learning
ER-Diagramme	ER-Modellierung	er-diagrams
Interaktiv		interactive
Labor, Übung		lab
Lehre		education
Lehrfach	Kurs, Lehrveranstaltung	course
Lernen		learning
Lernmaterial	Lehrmaterialien	learning assets, courseware
Normalisierung	BCNF	normalization, normalisation
Online	Web	online
Quiz		
Structured Query Language	SQL	
Tutor		tutor
Tutorial		tutorial
Unified Modeling Language	UML	
virtuell		virtual
Visualisierung		visualization
Werkzeug		tool

Tabelle 2.1: Suchbegriffe für die Recherche

Der Inhalt lässt sich, wie folgt, grob unterscheiden: Der Großteil der Quellen präsentiert interaktive Lehr- und Lernmaterialien und beschreibt deren Einsatz in der Lehre. Teilweise werden Erfahrungen geteilt und sogar Evaluationen durchgeführt, bei denen die Lernergebnisse der Studierenden untersucht wurden. In Einzelfällen wird auch auf die Konzeption, Entwicklung und Architektur der Materialien eingegangen. Außerdem wurden, bereits in der Lehre eingesetzte, Materialien nachträglich evaluiert. In weiteren Arbeiten werden unterschiedliche Materialien miteinander verglichen. Die Spalte „Kategorie“ ordnet den Inhalt der Veröffentlichungen entsprechend der folgenden Kategorien ein:

- Unter einer **Präsentation** wird das Vorstellen eines Lehr- oder Lernmaterials verstanden. Der Einsatz und die Benutzung der Materialien werden darin beschrieben. Außerdem wird ggf. auf Anforderungen und die Umsetzung eingegangen.
- In **Evaluationen** wird der Einsatz von interaktiven Materialien und deren Auswirkung auf die Lehre betrachtet. Dazu wurden Experimente mit Studie-

renden durchgeführt und sowohl Feedback eingeholt als auch der Lernerfolg untersucht.

- Ein **Vergleich** beschreibt wissenschaftliche Veröffentlichungen, die mehrere Materialien miteinander vergleichen; Ähnlich wie in dieser Arbeit.

Der Themenbereich, den die Veröffentlichung bzw. die Materialien umfasst, wird in der Spalte „Themenbereich“ angegeben. Die Themenbereiche wurden an, aus in der Lehre häufig referenzierten Fachliteratur *Datenbanksysteme: Eine Einführung* [KE15] und *Taschenbuch Datenbanken* [Kud15], orientiert. Somit soll eine einheitliche und schlüssige Einordnung gewährleistet werden.

Die Spalte „Veröffentlichung“ beschreibt entweder die Institution, das Journal oder die Konferenz, in der die Quelle veröffentlicht wurde. Die Anzahl der Zitierungen durch andere Quellen wurde mithilfe von Google Scholar¹² ermittelt und wird in der gleichnamigen Spalte aufgezählt.

2.2.4 Auswahlgründe

Die Veröffentlichungen wurden anhand mehrerer Kriterien hinsichtlich ihrer Wissenschaftlichkeit und Bedeutung für diese Arbeit untersucht. Die genauen Auswahlgründe der jeweiligen Quellen wurden in Tabelle 2.2 vermerkt. Im Folgenden wird genauer auf die Kriterien eingegangen.

Ein besonderer Fokus liegt auf der Verwendung freier und verfügbarer Materialien. Die Veröffentlichungen, die solche untersuchten, wurden bei der Auswahl bevorzugt. Dadurch kann nicht nur das Ergebnis besser nachvollzogen werden, sondern die Materialien können auch besser miteinander verglichen werden.

Sofern Evaluationen durchgeführt wurden, wurden diese besonders berücksichtigt. Wie viele Studierende teilgenommen haben und über welchen Zeitraum die Auswertungen durchgeführt wurden, spielen eine wichtige Rolle. Ebenfalls von Priorität waren jene Auswertungen, die nicht nur Studierende befragten, sondern auch deren Leistung (z. B. anhand von Noten) verglichen. Somit können genauere Annahmen über die Nützlichkeit und den Lernerfolg durch die Materialien getroffen werden.

Manche Themen werden nur von einer geringen Anzahl an Quellen behandelt. Um das Ziel der Recherche zu erfüllen (vgl. Abschnitt 2.2) wurde darauf geachtet, mit der Auswahl möglichst viele Themenbereiche abzudecken. Weitere Merkmale, die hinzugezogen wurden, sind das Jahr der Veröffentlichung und die Anzahl der Zitierungen.

¹²<https://scholar.google.com>

Tabelle 2.2: Ergebnis der Literaturrecherche

Lfd. Nr.	Auswahlgründe	Kategorie	Themenbereiche	Autoren	Titel	Veröffentlichung	Jahr	Zitierungen
1	AutoER; Automatisierte Aufgabenerstellung; Ansatz ähnlich von KERMIT nur für UML; Kürzliche Vorstellung.	Präsentation, Evaluation	UML	Sarah Foss, Tatiana Urazova, Ramon Lawrence	Automatic Generation and Marking of UML Database Design Diagrams	SIGCSE 2022: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1	2022	4
2	RelaX; Beschreiben den Einsatz des Materials in der Lehre	Präsentation	RA	Günther Specht, Johannes Kessler, Maximilian Mayerl, Michael Tschuggnall	RelaX	Datenbank-Spektrum volume 21, pages49-54 (2021)	2021	1
3	Gruppiert und vergleicht SQL Tools und evaluiert eine neue Methode für die Lehre	Vergleich, Evaluation	SQL	Shin-Shing Shin	Structured Query Language Learning: Concept Map-Based Instruction Based on Cognitive Load Theory	IEEE Access	2020	11
4	MonstER Park; Gamification als alternativer Ansatz; Einfache Bedienung und Aufbau.	Präsentation, Evaluation	ERM	Johannes Schildgen	MonstER Park - The Entity-Relationship-Diagram Learning Game	ER Forum, Demo and Posters 2020	2020	7
5	SQL-Tutor; zeigt die kontinuierliche Weiterentwicklung;	Präsentation, Evaluation	SQL	Faiza Tahair, Antonija Mitrovic & Valerie Sotardi	Investigating the Effects of Gamifying SQL-Tutor	Proceedings of the 28th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education	2020	15
6	SQL-Trainer; erweiteres Feedbacksystem mit einem Entscheidungsbaum	Präsentation	SQL	Faeskorn-Woyke, Heide; Bertelsmeier, Birgit; Strohschein, Jan	A Decision Tree Approach for the Classification of Mistakes of Students Learning SQL, a case study about SELECT statements	DELFTI 2020 – Die 18. Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V.	2020	3
7	Normalization Educational Tool; Desktop-Anwendung; Speziell BCNF.	Präsentation	NF	Christos Stefanidis, Georgia Koloniari	An interactive tool for teaching and learning database normalization	PCI '16: Proceedings of the 20th Pan-Hellenic Conference on Informatics	2016	5
8	EER-Tutor; Vergleich von bereits eingesetzten Materialien; Weiterentwicklungen von KERMIT.	Vergleich	ERM	Antonija Mitrovic, Pramuditha Suraweera	Teaching Database Design with Constraint-Based Tutors	International Journal of Artificial Intelligence in Education	2015	18
9	UML für ACME-DB; Integration in bestehendes LMS; Automatische Benotung.	Präsentation, Evaluation	UML	J. Soler, I. Boada, F. Prados, J. Poch, R. Fabregat	A web-based e-learning tool for UML class diagrams	IEEE EDUCON 2010 Conference	2010	64
10	SQL-Tutor; Zusammenfassung mehrerer Evaluationen	Präsentation, Evaluation	SQL	Antonija Mitrovic	An Intelligent SQL Tutor on the Web	International Journal of Artificial Intelligence in Education, vol. 13, no. 2-4, pp. 173-197, 2003	2003	458
11	NORMIT; Eine der ersten Materialien zur Normalisierung; Neuer Lernansatz (self-explanation); CBM Anwendung.	Präsentation, Evaluation	NF	Antonija Mitrovic	Supporting Self-Explanation in a Data Normalization Tutor	University of Canterbury	2003	37
12	KERMIT; Eines der ersten Materialien; Bedeutung für zukünftige Entwicklungen; Erprobung von CBM für ERM.	Präsentation, Evaluation	ERM	Pramuditha Suraweera, Antonija Mitrovic	KERMIT: A Constraint-Based Tutor for Database Modeling	International Conference on Intelligent Tutoring Systems - ITS 2002: Intelligent Tutoring Systems pp 377-387	2002	186
13	SQL-Tutor; historisch relevant	Präsentation, Evaluation	SQL	Antonija Mitrovic	Learning SQL with a computerized tutor	SIGCSE '98: Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education March 1998 Pages 307-311	1998	135

2.3 Interaktive Lehr- und Lernmaterialien

Im Gegensatz zur Recherche der wissenschaftlichen Veröffentlichungen gestaltet sich die Suche nach interaktiven Lehr- und Lernmaterialien dahingehend anders, dass andere Suchmaschinen und Portale verwendet wurden. In Unterabschnitt 2.3.1 werden diese näher beschrieben. Ziel ist es verfügbare Materialien zu finden und zu sammeln.

Zuerst wurden jedoch die Materialien aufgenommen, die aus den wissenschaftlichen Veröffentlichungen, aus der vorherigen Recherche, hervorgegangen sind. Falls die Materialien nicht direkt verlinkt waren, wurde über die Suchmaschinen online explizit nach den Projektnamen gesucht. Falls keine Materialien gefunden werden konnten, wurden die Namen der Autoren und Institutionen in der Suche integriert.

Danach wurde die Suche mithilfe der bereits beschriebenen Suchparameter (siehe Unterabschnitt 2.2.2) ausgeweitet. Auch hier wurde wieder eine rekursive Suche durchgeführt. Unter anderem wurde auch direkt auf den Profilen der Autoren auf den entsprechenden Portalen nach weiteren Publikationen gesucht.

Ebenfalls einbezogen wurden erweiterte Themengebiete, wie z. B. Datenstrukturen und Algorithmen, die in DSE zusätzlich gelehrt werden. Nicht berücksichtigt wurden sog. Computer-aided software engineering (CASE). Das sind Anwendungen und Werkzeuge, die in der Softwareentwicklung eingesetzt werden. Ihr Ziel ist es den Entwicklungs-Prozess zu vereinfachen oder zu beschleunigen und nicht inhaltlich auf die Anwendungsbereiche einzugehen. Daher sind sie in der Lehre nur bedingt von Nutzen. Hierunter fallen im allgemeinen Integrated Development Environments (IDEs), Datenbank-Modellierungswerkzeuge und Administrations-Werkzeuge. Beispiele sind die *MySQL Workbench*¹³, *SQLFiddle*¹⁴ und *PHPMyAdmin*¹⁵.

2.3.1 Portale

Es wurden andere Portale und Suchmaschinen verwendet, um Materialien, die meistens nur online verfügbar sind und nicht als Teil von wissenschaftlichen Arbeiten veröffentlicht wurden, zu erfassen.

Konventionelle Suchmaschinen, wie z. B. Google¹⁶ und DuckDuckGo¹⁷, lieferten ein

¹³Produktseite der MySQL Workbench: <https://www.mysql.com/products/workbench/>

¹⁴<http://sqlfiddle.com/>

¹⁵Dokumentation von PHPMyAdmin: <https://docs.phpmyadmin.net/en/latest/intro.html>

¹⁶<https://google.com>

¹⁷<https://duckduckgo.com/>

breites Spektrum an Ergebnissen. Daneben wurde GitHub¹⁸, eine der führenden kollaborativen Plattformen für Open-Source-Software, verwendet. Der Quellcode vieler Materialien ist dort, teilweise unter freier Lizenz, veröffentlicht.

Auf Open Educational Resources (OER) spezialisierte Suchmaschinen unterstützen die Suche, indem sie optimierte Suchparameter an allgemeine Suchmaschinen übergeben. So auch „OERHörnchen“¹⁹, welches eine Suchanfrage an Google vorbereitet. Gleichzeitig sammeln OER-Datenbanken, wie etwa „Open Resources Campus NRW (ORCA)“²⁰, frei verfügbaren Materialien. Beide wurden für die Recherche verwendet.

2.3.2 Ergebnis

Basierend auf dem oben beschriebenen Vorgehen wurden die in Tabelle 2.3 aufgelisteten Materialien gefunden. Um eine detaillierte Übersicht der Materialien zu erzeugen, wurden einzelne Merkmale erarbeitet, die im Folgenden näher erläutert werden.

Die Materialien unterscheiden sich signifikant in ihrer Methodik, Inhalte zu vermitteln. Die zweite Spalte „Methode“ hebt diese Unterschiede hervor. Die folgenden Methoden wurden erarbeitet:

- Eine **Übung** beschreibt interaktive Aufgaben, bei denen Studierende selbstständig Lösungen erarbeiten und einreichen müssen.
- Unter **Quiz** werden Single- und Multiple-Choice Aufgaben, Kreuzworträtsel, Textantworten, etc. verstanden.
- Mit **Theorie** gekennzeichneten Materialien werden Themen erklärt, Konzepte eingeleitet oder Hilfestellungen gegeben.
- **Visualisierungen** stellen komplexe Algorithmen und Mechanismen grafisch dar.

Die Materialien wurden auch hier wieder in die in Unterabschnitt 2.2.3 bereits vorgestellten Themenbereichen eingeordnet. Einzelne Materialien können auch mehrere Themenbereiche umfassen.

Ein wichtiges Merkmal veröffentlichter Materialien ist die Lizenz, unter der diese verwendet, verändert und verbreitet werden dürfen. Abhängig von der Umsetzung wurden sowohl Quellcode als auch die erarbeiteten Inhalte veröffentlicht. Beide

¹⁸<https://github.com>

¹⁹<https://oerhoernchen.de/>

²⁰<https://www.orca.nrw/>

werden meistens durch unterschiedliche Lizenzen bereitgestellt. Sofern ersichtlich, wurden die Lizenzen in der Spalte „Lizenz“ angegeben. Andernfalls wurden sie mit „unlicensed“ markiert.

In der Spalte „Umsetzung“ wird beschrieben, wie die Materialien implementiert wurden und v. a. wie mit ihnen interagiert werden kann. Materialien, die mit „Web“ gekennzeichnet sind, können in einem Webbrowser aufgerufen werden. Hingegen werden Materialien, die mit dem Label „Desktop“ versehen sind, direkt auf dem Endgerät ausgeführt.

Die vorletzte Spalte „Zugriff“ gibt an in welcher Form die Materialien zum Zeitpunkt der Recherche zur Verfügung stehen. Sofern zur Benutzung eine Anmeldung in einer entsprechenden Umgebung nötig ist, wurde dies mit „eingeschränkt“ vermerkt. Nicht mehr verfügbare Materialien sind als „nicht verfügbar“ gekennzeichnet.

In der letzten Spalte „Link“ wird auf das verfügbare Material verlinkt. Die Verlinkung führt entweder direkt zur online verfügbaren Version, andernfalls zu der Projektseite. Ansonsten wurde der Quellcode der Software referenziert.

2.3.3 Auswertung

Anhand der in Unterabschnitt 2.3.2 erläuterten Merkmale wird in diesem Abschnitt das Ergebnis der Materialrecherche eingeordnet. Unter anderem wird auch auf die quantitativen Besonderheiten der Merkmale in Tabelle 2.3 eingegangen.

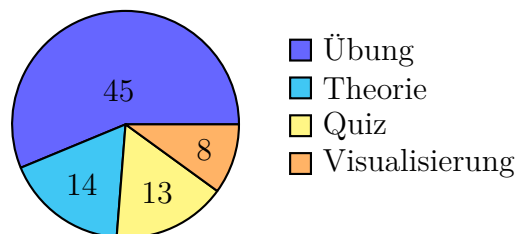


Abbildung 2.2: Übersicht über die Methodiken

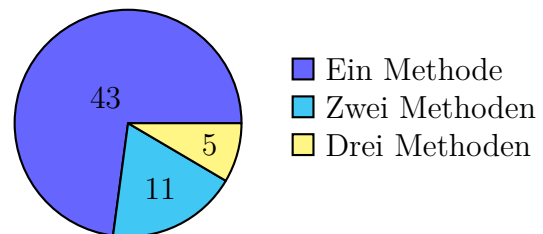


Abbildung 2.3: Anzahl an Methodiken pro Material

Aus der Aufteilung der Materialien hinsichtlich der Methodik ergibt sich Abbildung 2.2. Die überwiegende Mehrheit der Materialien kann durch „Übungen“ beschrieben werden. Darauf folgen mit jeweils nur einem geringen Anteil „Quiz“, „Theorie“ und „Visualisierungen“. Wichtig zu beachten ist jedoch, dass manche Materialien mehrere Methoden in Kombination einsetzen. Dies zeigt Abbildung 2.3 auf.

Aus der Recherche können vier Hauptthemenbereiche abgeleitet werden (vgl. Abbildung 2.4). Der Themenbereich „SQL“ ist am häufigsten in der Recherche vertreten, gefolgt von „ERM“ (Entity-Relationship-Modell), „NF“ (Normalform/Normalisierung) und „RA“ (Relationale Algebra). 78% der recherchierten Materialien sind spezialisierte Materialien, welche nur einen Themenbereich abdecken (vgl. Abbildung 2.5).

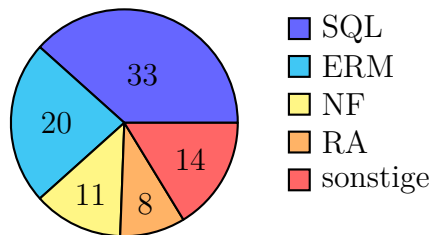


Abbildung 2.4: Übersicht über die Themenbereiche

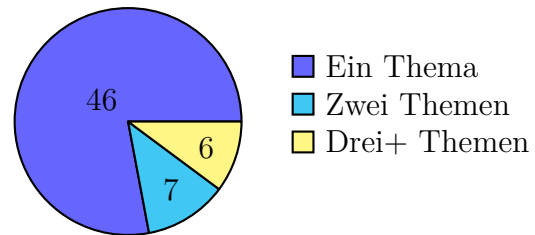


Abbildung 2.5: Anzahl an Themen pro Material

In Abbildung 2.6 wird die Verteilung der genutzten Lizenzen dargestellt. Insbesondere ist auffällig, dass zum Zeitpunkt der Recherche ein Großteil der Materialien (37 von 59) ohne jegliche Lizenzen veröffentlicht wurde. In diesen Fällen ist also unklar, wie diese verwertet werden dürfen. Ebenso problematisch ist die Auswahl von Softwarelizenzen für Lehrinhalte oder Creative Commons Lizenzen für Software. Von letzter wird abgeraten, da die Distribution von Quellcode nicht geregelt sei²¹. Der ER-REL-Trainer umgeht dieses Problem, indem Inhalt und Quellcode unter verschiedenen offenen Lizenzen veröffentlicht wurden²².

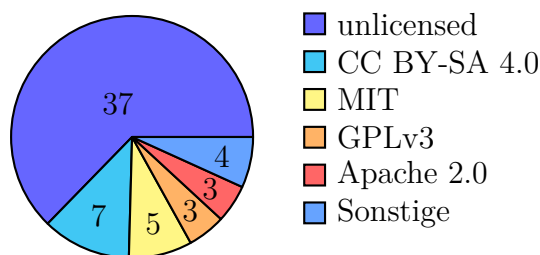


Abbildung 2.6: Lizenzen in den Materialien

²¹<https://creativecommons.org/faq/#can-i-apply-a-creative-commons-license-to-software>, aufgerufen am 17.04.2023

²²siehe Projektdokumentation https://github.com/EILD-nrw/er_rel_trainer#lizenzen, aufgerufen am 17.04.2023

Der Großteil der Materialien wurde als Webanwendungen realisiert und kann direkt online abgerufen werden. Vereinzelt werden Materialien auch als Desktopanwendungen, bspw. in Form von Java Applikationen, bereitgestellt. In seltenen Fällen erfordern Übungen die Installation zusätzlicher Software (z. B. Docker) auf den Endgeräten der Benutzer (siehe Tabelle 2.3).

Tabelle 2.3: Ergebnis der Materialrecherche

Lfd. Nr.	Methode	Themenbereiche	Autoren	Institution	Titel	Lizenz	Umsetzung	Zugriff	Link
1	Übung	ERM, NF, RA, SQL	F. Prados, I. Boada, J. Soler, J. Poch	Universitat de Girona, Spain	ACME-DB	unlicensed	Web	nicht verfügbar	
2	Übung	ERM, SQL	Meg Murray, Mario Guimaraes	Kennesaw State University	ADbC	unlicensed	Web, Desktop	nicht verfügbar	
3	Übung, Quiz, Visualisierung	ERM, SQL, NF	Kerstin Schneider, Fabian Keller, Pia Habekost, Victor Schmeil, Markus Dampmann, Florian Schorch	Hochschule Harz	ALEA – Adaptive eLEARNING System	unlicensed	Web	nicht verfügbar	
4	Übung	Transaktionsverwaltung	Alexander Kosmehl	EILD.nrw	Anomalie-Trainer	MIT	Web	frei	https://eild-nrw.github.io/anomaly_trainer/
5	Übung	ERM	Ramon Lawrence	The University of British Columbia	AutoER	BSD 3-Clause	Web	eingeschränkt	https://github.com/rlawrenc/autoer
6	Visualisierung	Datenstrukturen	Jonas Baur, Charlotte Hasler	EILD.nrw	B-Baum-Animation	MIT, CC BY-SA 4.0	Web	frei	https://eild-nrw.github.io/btree-animate/
7	Visualisierung	Datenstrukturen	Alexander Kosmehl	EILD.nrw	B-Baum-Zeichner	CC BY-SA 4.0	Web	frei	https://eild-nrw.github.io/b-tree/
8	Visualisierung	Datenstrukturen, Algorithmen	David Galles	University of San Francisco	Data Structure Visualizations	unlicensed	Web	frei	https://www.cs.usfca.edu/~sim\$galles/visualization/Algorithms.html
9	Übung, Theorie	ERM, SQL		serlo.org	Datenbanken und Datenbanksysteme	CC BY-SA 4.0	Web	frei	https://de.serlo.org/informatik/69501/datenbanken-und-datenbanksysteme
10	Quiz	SQL	Alexander Kosmehl	EILD.nrw	Datenbank-Kreuzwörterrätsel	CC BY-SA 4.0	Web	frei	https://eild-nrw.github.io/crossword/
11	Übung	NF	David Becher	Technische Universität München	DB->normalizer	unlicensed	Web	frei	https://normalizer.db.in.tum.de/
12	Übung	SQL, ERM, NF	Srinual Nalintippayawong, Kanokwan Atchariyachanvanich, Thanakrit Julavanich	KMITL	DBLearn	unlicensed	Web	eingeschränkt	https://dblearning.it.kmitl.ac.th/
13	Übung	SQL, ERM	Vangel V. Ajanovski		dbLearn*	GPLv3	Web	frei	https://github.com/ajanovski/dbLearnStar
14	Quiz	SQL, ERM, NF, RA, Anwendungsprogrammierung, Transaktionsverwaltung		TH-Köln	DB-Puzzle (SQL-Puzzle)	unlicensed	Web	frei	https://edb2.gm.th-koeln.de/apps/sqlpuzzle/
15	Quiz	NF, SQL, ERM, RA, Transaktionsverwaltung	Dr Gordon Russell	Edinburgh Napier University	DMBS Implementation - Online Quiz	unlicensed	Web	frei	https://db.grussell.org/mc/
16	Übung, Theorie	SQL	Andreas Thor, Toralf Kirsten	HTWK Leipzig, Hochschule Mittweida	E-Assessment-Tool DMT	Apache 2.0	Web	nur Quellcode	https://github.com/andreas-thor/dasp-dmt
17	Übung	ERM	Pramuditha Suraweera, Antonija Mitrovic	University of Canterbury	EER-Tutor	unlicensed	Web	eingeschränkt	

Tabelle 2.3 – Fortsetzung von vorheriger Seite

Lfd. Nr.	Methode	Themenbereiche	Autoren	Institution	Titel	Lizenz	Umsetzung	Zugriff	Link
18	Übung, Theorie	ERM	Pramuditha Suraweera, Antonija Mitrovic und Weitere	University of Canterbury	EER-Tutor (Web-enabled KERMIT)	unlicensed	Web	eingeschränkt	https://ictg.cosc.canterbury.ac.nz:8005/eer-tutor/login
19	Übung	ERM	André Kless	EILD.nrw	ER-REL-Trainer	MIT, CC0 1.0	Web	frei	https://eild-nrw.github.io/er_rel_trainer/
20	Quiz	ERM		TH-Köln	ER-Trainer	unlicensed	Web	eingeschränkt	https://edb2.gm.th-koeln.de/apps/ertrainer/
21	Übung	SQL	Aleš Cepek, Jan Pytel, Martin Landa	GNU	GNU Sqtutor	GPLv3	Web	frei	https://sqltutor.fsv.cvut.cz/cgi-bin/sqltutor
22	Visualisierung	RA	Henrik Mühe	Technische Universität München	IRA - Interaktive RA	unlicensed	Web	frei	https://db.in.tum.de/~simsmuehe/ira/
23	Übung	ERM	Pramuditha Suraweera, Antonija Mitrovic	University of Canterbury	KERMIT	unlicensed	Desktop	nicht verfügbar	
24	Übung	ERM	Pramuditha Suraweera, Antonija Mitrovic	University of Canterbury	KERMIT	unlicensed	Desktop	nicht verfügbar	https://www.csse.canterbury.ac.nz/tanja.mitrovic/kermit.html
25	Übung	NF	Nikolay Georgiev	University of Umeå	LDBN - Learn Database Normalization	Apache 2.0	Web	nur Quellcode	https://github.com/devng/ldbn
26	Übung, Theorie	SQL		learnsqloonline.org	Learn SQL Online	Apache 2.0	Web	frei	https://www.learnsqloonline.org/
27	Übung	ERM	Olav Dæhli, Bjørn Kristoffersen, Tomas Sandnes	University of South-Eastern Norway	LearnER	unlicensed	Web, Desktop	nicht verfügbar	
28	Übung	ERM	André Kless	EILD.nrw	LOS-Trainer	MIT	Web	frei	https://github.com/EILD-nrw/los_trainer
29	Übung, Theorie, Quiz	SQL	Böhme, T., Rahm, E. und Sosna, D.	Universität Leipzig	LOTS - Leipzig online test system	unlicensed	Web	eingeschränkt	https://lots.uni-leipzig.de/home/
30	Quiz	ERM, RA, NF, SQL, Anwendungsprogrammierung	Alexander Kosmehl	EILD.nrw	MCT-Trainer	CC BY-SA 4.0	Web	frei	https://eild-nrw.github.io/mct-trainer/
31	Übung	ERM	Johannes Schildgen	University of Applied Sciences in Regensburg	MonstER Park	unlicensed	Web	frei	https://www.monst-er.de/
32	Übung	NF	Alexander Kosmehl	EILD.nrw	NF-Trainer	CC BY-SA 4.0	Web	frei	https://eild-nrw.github.io/nf-trainer/
33	Übung	NF	Christos Stefanidis, Georgia Koloniari	University of Macedonia	Normalization Education Tool	unlicensed	Desktop	frei	http://users.uom.gr/~sim\$gkoloniari/normTool.html
34	Übung, Theorie	NF	Antonija Mitrovic	University of Canterbury	NORMIT	unlicensed	Web	nicht verfügbar	https://www.csse.canterbury.ac.nz/tanja.mitrovic/normit.html
35	Theorie, Quiz	Anwendungsprogrammierung	Arabella Jackszis, Melina Eleftheriadis	Hochschule Düsseldorf	Objektrelationales Mapping in Datenbanksystemen	unlicensed	Web	frei	https://h5p.org/h5p/embed/1263725
36	Quiz	ERM	Pete Thomas	Open University	OpenMark - Online Automatic Marking of Diagrams	GPLv3	Web	eingeschränkt	https://www.open.ac.uk/openmarkexamples/

Tabelle 2.3 – Fortsetzung von vorheriger Seite

Lfd. Nr.	Methode	Themenbereiche	Autoren	Institution	Titel	Lizenz	Umsetzung	Zugriff	Link
37	Übung	SQL		TH-Köln	PL/SQL-Trainer	unlicensed	Web	frei	https://edb2.gm.th-koeln.de/apps/plsql/
38	Übung	RA, SQL	Christina Ehrlinger, Thomas Fritsch, Michael Fruth, Franz Lehner, Stefanie Scherzinger	Universität Passau	Praktomat für RADB und PSQL	unlicensed	Web	nur Quellcode	https://github.com/sdbs-uni-p/praktomat-for-radb-and-psql
39	Übung	SQL	Kristofer Koishigawa, Tom Mondloch	freeCodeCamp	Relational Database Course	unlicensed	VSCoDe & Docker	frei	https://www.freecodecamp.org/news/how-to-run-freecodecamps-relational-databases-curriculum-using-docker-vscode-and-coderoad/
40	Visualisierung	RA	Günther Specht, Johannes Kessler, Maximilian Mayerl, Michael Tschuggnall	Forschungsgruppe DBIS - Universität Innsbruck	RelaX – Interaktive RA in der Lehre	unlicensed	Web	frei	https://dbis-uibk.github.io/relax/landing
41	Übung, Theorie	Transaktionsverwaltung	André Kless	EILD.nrw	Serialization-Trainer	MIT	Web	frei	https://eild-nrw.github.io/serialization_trainer/
42	Übung, Theorie	SQL	Johannes Schildgen, Stefan DeBloch	Technischen Universität Kaiserslautern	SQL Island	unlicensed	Web	frei	https://sql-island.informatik.uni-kl.de/
43	Übung	SQL	Anthony Kleerekoper and Andrew Schofield	Manchester Metropolitan University	SQL tester	MIT	Web	nur Quellcode	https://github.com/kleerekoper/SQLTester
44	Übung, Theorie	SQL		SQLBolt	SQLBolt	unlicensed	Web	frei	https://sqlbolt.com/
45	Übung	SQL	Christoph Gerstle, Florian Moritz, Bernhard Schiefer	Hochschule Kaiserslautern	SQLcoach	LGPL	Web	frei	https://sqlcoach.informatik.hs-kl.de/sqlcoach/
46	Übung, Theorie, Quiz	SQL		TechnologyAdvice	SQLcourse	unlicensed	Web	frei	https://www.sqlcourse.com/
47	Übung	SQL	Kamil Akhuseyinoglu	PAWS Lab at University of Pittsburgh	SQL-KnoT	unlicensed	Web	nur Quellcode	https://github.com/PAWSLabUniversityOfPittsburgh/sql_knot
48	Übung	SQL	Anthony Kleerekoper	Manchester Metropolitan University	SQLTester	MIT	Web	nur Quellcode	https://github.com/kleerekoper/SQLTester
49	Übung	SQL	Alexander Kosmehl	EILD.nrw	SQL-Trainer	CC BY-SA 4.0	Web	frei	https://eild-nrw.github.io/sql-trainer/
50	Übung	SQL	Dieter Sosna, Erhard Rahm	Universität Leipzig	SQL-Trainer	unlicensed	Web	nicht verfügbar	http://sql-trainer.uni-leipzig.de/
51	Übung	SQL	Harri Laine	University of Helsinki	SQL-trainer	unlicensed	Web	eingeschränkt	https://www.cs.helsinki.fi/u/laine/tikape/trainer.html
52	Übung	SQL	Antonija Mitrovic, Kurt Hausler	University of Canterbury	SQL-Tutor (SQLT-Web)	unlicensed	Web, Desktop	eingeschränkt	https://ictg.cosc.canterbury.ac.nz:8000/sql-tutor/login
53	Übung	SQL	Victor Obionwu	Universität Magdeburg	SQLValidator	unlicensed	Web	eingeschränkt	https://propra14.iti.cs.ovgu.de/sqlValidator/publicRoot/sqlvali/index.php

Tabelle 2.3 – Fortsetzung von vorheriger Seite

Lfd. Nr.	Methode	Themenbereiche	Autoren	Institution	Titel	Lizenz	Umsetzung	Zugriff	Link
54	Übung, Theorie	SQL	Andrew Cumming		SQLZoo	unlicensed	Web	frei	https://sqlzoo.net/wiki/SQL_Tutorial
55	Theorie, Quiz	UML	Thomas Rakow & Mareike Focken	Hochschule Düsseldorf	UML Klassenmodellierung	CC BY-SA 4.0	Web	frei	https://www.orca.nrw/content/6cc17335-a639-4d0d-a7e5-c1780eed83fe
56	Visualisierung	Datenstrukturen, Algorithmen	Dr Steven Halim	National University of Singapore	VisuAlgo	unlicensed	Web	frei	https://visualgo.net/en
57	Theorie, Übung, Quiz	SQL		W3Schools	W3Schools SQL	unlicensed	Web	frei	https://www.w3schools.com/sql
58	Übung, Quiz, Visualisierung	SQL	Aaron Paul M. Dela Rosa, Luigi Miguel M. Villanueva, John Mardy R. San Miguel, John Emmanuel B. Quinto	Bulacan State University	Web-based Database Courses E-Learning Application	unlicensed	Web	nicht verfügbar	
59	Übung	RA, SQL	Suzanne W. Dietrich, Eric Eckert, Kevin Piscator	Arizona State University	WinRDBI Educational Tool	unlicensed	Desktop	frei	https://winrdbi.asu.edu/

2.4 Sonstige

Während des gesamten Entwicklungsprozesses einer Datenbank bis hin zur Administration kommen häufig komplexe Software und Technologien zum Einsatz. Teilweise sind diese proprietär und herstellereinspezifisch. Weil der Einsatz dieser Software nicht unbedingt trivial ist, bieten deren Hersteller häufig eigene Lehr- und Lernmaterialien an. Ein Beispiel ist die Oracle University²³, die Lernende an die Oracle Produkte heranführen soll. Inhaltlich wird in den Materialien auch auf Themen der DSE eingegangen, schließlich wurde die Software dafür entwickelt.

Daher wären solche Angebote ebenfalls für diese Recherche relevant. Allerdings wurden sie aus den folgenden Gründen nicht berücksichtigt: 1) Die Lernmaterialien sind spezifisch für ein bestimmtes Produkt oder einen bestimmten Hersteller entwickelt worden. Auf Grund von Unterschieden zu anderen Produkten oder der DSE können dessen Inhalte nicht verallgemeinert werden. 2) Die Angebote sind teilweise nicht frei verfügbar, sondern erfordern eine Anmeldung beim jeweiligen Hersteller oder können sogar an kommerzielle Lizenzen gebunden sein. 3) Es liegt im Ermessen der Hersteller das Angebot jederzeit einzustellen oder zu verändern. Da keine freien Lizenzen verwendet werden, dürfen Dritte die Inhalte i. d. R. weder anpassen noch selbst verbreiten.

2.4.1 Onlinekurse

Online wird eine Vielzahl an Kursen im Bereich der Datenbanksysteme von Dritten auf Plattformen wie Udemy²⁴, Hyperskill²⁵, edX²⁶ und vielen Weiteren angeboten. Diese Kurse sind allerdings nicht unbedingt interaktiv, bzw. nur insofern, dass Inhalte und praktischen Übungen angeboten werden. Auf der anderen Seite bieten manche Plattformen online Code-Editoren oder Integrationen an, die ein interaktives Lernen ermöglichen. Im Allgemeinen wurden diese Angebote aus ähnlichen Gründen ebenfalls nicht berücksichtigt: 1) Die Angebote sind häufig kostenpflichtig und daher nicht zugänglich. 2) Ebenso stehen sie unter keiner freien Lizenz. 3) Im Zweifel kann die Qualität der Inhalte nicht verifiziert werden.

²³Angebote in der Oracle University: <https://education.oracle.com/learning-explorer#startLearning>, aufgerufen am 01.05.2023

²⁴Suchergebnis von „database systems“ auf Udemy: <https://www.udemy.com/courses/search/?src=ukw&q=database+systems>, aufgerufen am 01.05.2023

²⁵Angebote in der Kategorie SQL auf Hyperskill: <https://hyperskill.org/tracks?category=9>, aufgerufen am 01.05.2023

²⁶Inhalte im Bereich von Datenbanken auf edX: <https://www.edx.org/learn/databases>, aufgerufen am 01.05.2023

Kapitel 3

Ergebnisse

In diesem Teil der Arbeit werden die wissenschaftlichen Veröffentlichungen und die interaktiven Lehr- und Lernmaterialien inhaltlich ausgewertet und näher beschrieben. Das Kapitel ist in zwei Abschnitte aufgeteilt. Der Fokus liegt auf der Beschreibung einzelner Materialien in Abschnitt 3.2. Zunächst werden allgemeine wissenschaftliche Veröffentlichungen in Abschnitt 3.1 zusammengefasst.

3.1 Wissenschaftliche Arbeiten

Die überwiegende Mehrheit der wissenschaftlichen Veröffentlichungen befasst sich direkt mit interaktiven Materialien. Diese wurden oft begleitend zu den Arbeiten entwickelt oder im Anschluss evaluiert. Weiter wurden Studien und wissenschaftliche Arbeiten erfasst, die mehrere Materialien miteinander vergleichen, oder deren Einsatz in der Lehre bewerten. Letztere werden in diesem Abschnitt vorgestellt.

Advances in database systems education In [Ish+22] analysieren Ishaq et. al. insgesamt 70 wissenschaftliche Publikationen, die sich mit der Lehre von Datenbanksystemen (DSE) befassen [siehe Ish+22, S. 2686 f.]. Mit ihrerer systematischen Übersichtsarbeit möchten die Autoren den aktuellen Stand der Forschung zur DSE erfassen und dadurch eine Diskussionsgrundlage für zukünftige Anforderungen und Entwicklungen in der DSE schaffen [vgl. Ish+22, S. 2684–2686]. Die Recherche wurde anhand von zuvor festgelegten Zielen und Fragestellungen durchgeführt. Die ausgewählte Literatur wurde im Anschluss daran in die drei Kategorien „Werkzeuge“, „Methodik“ und „Studienplan“ (engl. „Curriculum“) aufgeteilt.

Unter „Werkzeuge“ verstehen die Autoren Anwendungen, die in der Lehre eingesetzt werden können, um Themen besser vermitteln zu können [vgl. Ish+22, S. 2690]. Seit

2004 ist in dieser Kategorie ein großer Anstieg an Veröffentlichungen zu erkennen [vgl. Ish+22, S. 2711]. Zum einen wurden dedizierte Werkzeuge erfasst, die für spezifische Themenbereiche und Aufgabenstellungen entwickelt wurden; Also genau solche, die auch in Abschnitt 3.2 vorgestellt werden. Im Allgemeinen stellen die Autoren fest, dass der Fokus dabei auf Lernmethoden für Studierende liegt [siehe Ish+22, S. 2694]. Auf der anderen Seite werden Werkzeuge beschrieben, die dafür entworfen wurden, den klassischen Unterricht digital zu gestalten. Über Kommunikationstechnologien (Chats), digitale Textbücher und interaktive Whiteboards sollen Studierende auch online unterrichtet werden können. Diese Form des Unterrichts würde immer mehr an Bedeutung gewinnen, nicht zuletzt wegen der COVID Pandemie [siehe Ish+22, S. 2715]. Ebenfalls sei die Individualisierbarkeit der Werkzeuge an Studierende und der engere Austausch zwischen Lehrende und Lernende auf Plattformen von Vorteil [siehe Ish+22, S. 2695].

Die Kategorie „Methodik“ beschreibt Lehr- und Lernmethoden, die angewendet werden, um das Wissen zu vermitteln. Auch hier konnten Fortschritte festgestellt werden. Verschiedene Methoden wurden in mehreren wissenschaftlichen Arbeiten erprobt [vgl. Ish+22, S. 2695]. Ein Grundinteresse ist Voraussetzung für erfolgreiches Lernen [vgl. Ish+22, S. 2699]. Durch interaktive Medien und praxisorientierte Probleme kann das Interesse der Studierenden an dem Themengebiet geweckt werden [siehe Ish+22, S. 2715]. Ein Kritikpunkt an aktuellen Lehrmethoden ist, dass primär theoretisches Wissen vermittelt wird. Um mit dem technologischen Fortschritt mithalten zu können, müsse auch praktische Erfahrungen gesammelt werden können [siehe Ish+22, S. 2700]. Allerdings wurden nur in 16% der Veröffentlichungen die angewandten Methoden evaluiert [siehe Ish+22, S. 2718].

Unter der dritten Kategorie fassen die Autoren alle Veröffentlichungen zusammen, die sich mit den Lehrinhalten, also dem Studienplan, beschäftigen. Für das Fach Datenbanken existieren Richtlinien und Standards, die Ziele, Inhalt und Umfang von Lehrplänen definieren [siehe Ish+22, S. 2705 f.]. Im Allgemeinen wurden diese Richtlinien im Verlauf der Zeit mehrfach überarbeitet [siehe Ish+22, S. 2705], um auch neuere Technologien miteinzubeziehen [siehe Ish+22, S. 2713].

Structured Query Language Learning: Concept Map-Based Instruction Based on Cognitive Load Theory Shin stellt eine neue Methode zum Erlernen der Datenbankabfragesprache SQL in seiner Veröffentlichung [Shi20b] vor und evaluiert diese. In der Literaturrecherche der Arbeit werden Materialien, die über die konventionelle Lehre hinausgehen, recherchiert und eingeordnet. Darüber hinaus wird in der Arbeit die neue Methodik auf ihre Effektivität hin in der Evaluation beurteilt.

Die recherchierten Materialien hat der Autor in drei Kategorien („animation“, „graphical query builder“ und „feedback“) gruppiert, zeigt die Schwierigkeiten bei dem Lernen mit den Kategorien auf und stellt einige Anwendungen im Detail vor [siehe Shi20b, S. 100096 f.].

Der Autor stellt fest, dass die kognitive Belastung der Studierenden ein zentrales Problem bei der Lehre von SQL darstellt [siehe Shi20b, S. 100095 f.]. Um dieses Problem anzugehen, hat Shin eine Lehrmethode entwickelt, die auf einer „Concept Map“ basiert und SQL-Abfragen illustriert. Das Ziel dieser Methode besteht darin, die kognitive Belastung zu reduzieren und somit bessere Lernergebnisse zu erzielen [vgl. Shi20b, S. 100095]. Die Grundlage für diese These bildet die „Cognitive Load Theory“.

Die Evaluation wurde in zwei Datenbankkursen mit 39 respektive 42 Studierenden durchgeführt. Dabei ist der zweite Kurs die Kontrollgruppe. In dem ersten Kurs wurde SQL in konventioneller Art erklärt. Hingegen wurde bei dem zweiten Kurs die „Concept map“ als primäres Mittel zur Erklärung eingesetzt. Es wurden Daten zur mentalen Anstrengung sowie zur Leistung bei der Lösung von SQL-Problemen der Studierenden erhoben. [vgl. Shi20b, S. 100103]. In der Auswertung der Ergebnisse wurde festgestellt, dass der Datenbankkurs mit der „Concept map“ eine geringere mentale Anstrengung benötigte als der mit der konventionellen. Aufgrund der besseren Effizienz ist daher von einem besseren Lernerfolg auszugehen [vgl. Shi20b, S. 100104 f.].

3.2 Materialien

Im Folgenden werden nun interaktive Lehr- und Lernmaterialien und begleitende wissenschaftliche Arbeiten vorgestellt. Dabei wird auf die Anforderungen, Umsetzung, Inhalte, Funktionsweise, und Evaluation, sofern möglich, eingegangen. Die Unterteilung der Abschnitte orientiert sich an den in Unterabschnitt 2.2.3 vorgestellten Themenbereichen.

3.2.1 Entity-Relationship-Modell

Das ERM ist ein relativ simples Modell zum Beschreiben von Objekten (engl. „entities“) und ihren Beziehungen untereinander. Es wird häufig zum Entwurf von Datenbanken eingesetzt, um ein aus den Anforderungen abgeleitetes Daten-Modell semantisch zu beschreiben. In einem Entity Relationship Diagramm (ER-Diagramm) werden die Modellelemente durch geometrische Formen dargestellt und über Linien miteinander verbunden. Erweiterungen des ERM werden auch als Erweitertes Entity-Relationship-Modell (EERM) bezeichnet [vgl. Kud15, S. 50, 60].

Kapitel 3. Ergebnisse

3.2. Materialien

Peter P. Chen stellte 1976 zwar Regeln auf, wie einzelne Komponenten dargestellt werden sollen, dennoch gibt es keine strikte Herangehensweise. Zudem kann es zu einer Aufgabenstellung mehrere korrekte Lösungen geben, die sich untereinander unterscheiden [vgl. SM02, S. 377; MS16, S. 449 f.]. Suraweera und Mitrovic bemängeln darüber hinaus häufig unvollständige oder ungenaue Aufgabenstellungen [vgl. SM02, S. 377].

Mithilfe der in diesem Abschnitt vorgestellten Materialien, sollen Studierende das Anfertigen von ERM schrittweise erlernen und üben.

KERMIT ist eine Desktopanwendung zum Erlernen von ERM. Studierende können in einer grafischen Oberfläche, mithilfe von Microsoft Visio, ER-Diagramme erstellen. Sie bekommen verschiedene Textaufgaben gestellt und müssen diese bearbeiten. Anschließend werden ihre Eingaben automatisiert ausgewertet und ggf. Feedback gegeben. KERMIT setzt dabei auf ein, von Suraweera und Mitrovic als „Constraint-Based Modeling (CBM)“ bezeichnetes, Verfahren zur Überprüfung der Eingaben [siehe SM02, S. 377 f.].

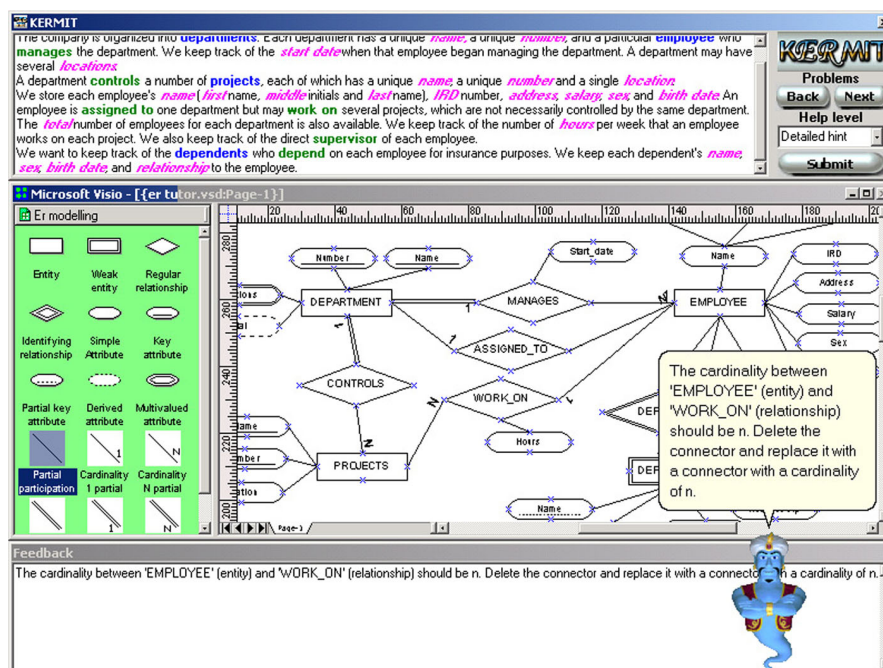


Abbildung 3.1: Screenshot zeigt die Bearbeitung einer Aufgabe in KERMIT [aus MS16, Abbildung 1.]

Weitere Materialien basieren auf KERMIT und erweitern dessen Funktionen [siehe SM02, S. 385 f.; MS16, S. 451–455]. e-KERMIT ist eine Weiterentwicklung und

ermöglicht die Visualisierung der Problemfelder und des Lernfortschrittes der Studierenden. Damit sollen sich Studierende besser selbst einschätzen können [vgl. MS16, S. 451 f.]. Mit dem EER-Tutor wurde KERMIT ins Web portiert. Außerdem wurde es um Funktionalitäten zum Erstellen von EER-Diagrammen erweitert. Die online Version ermöglicht einen einfacheren Zugang zu der Software. Dennoch steht sie nur Studierenden der „University of Canterbury“ über eine Anmeldung zur Verfügung.

Die Besonderheit beim Erstellen des Diagramms ist, dass Studierende erst Passagen oder Sätze im Text markieren müssen, bevor sie Elemente dem Diagramm hinzufügen können. So soll sichergestellt werden, dass die Studierenden den Aufgabentext sorgfältig durchgelesen und verstanden haben [vgl. SM02, S. 381 f.]. Gleichzeitig werden dadurch einheitliche Begriffe im Diagramm verwendet, was bei der Auswertung der Lösung helfen kann [vgl. SM02, S. 381 f.].

Abbildung 3.1 zeigt einen Screenshot der Anwendung. Im oberen Teil ist der Aufgabentext zu sehen. Hier werden, wie oben beschrieben, einzelne Wörter farblich hervorgehoben. Gleich darunter befindet sich das Microsoft Visio Fenster. Auf der linken Seite davon stehen den Studierenden alle Modellelemente zur Verfügung, die sie auf der rechten Seite zum Diagramm hinzufügen können. Im unteren Teil ist das Fenster mit dem Feedback zu erkennen. Zusätzlich wird ein Avatar (siehe unten rechts) dargestellt, der das Feedback nochmals in einer Sprachblase anzeigt aber zusätzlich auch auditiv wiedergibt.

Unter CBM beschreiben Suraweera und Mitrovic die Methode, die Eingabe mithilfe speziell einprogrammierter Bedingungen (engl. „constraints“) auszuwerten. Damit kann festgestellt werden, wie sehr die Eingabe mit der Musterlösung übereinstimmt und ob die Anforderungen erfüllt wurden.

Wenn eine Bedingung, z. B. dass die Namen der Objekte groß geschrieben werden sollen, nicht erfüllt wurde, kann daraus Feedback erzeugt werden. Das Feedback wird in mehrere Detailstufen aufgeteilt. Die größte Einstellung deutet lediglich an, ob die Bedingung erfüllt ist, also die Eingabe dahingehend korrekt ist. Eine „error flag“ gibt an, welches Element gemeint ist. Weitere Detailstufen können ausführlichere Hinweise oder sogar Lösungen geben. KERMIT ist so konfiguriert, dass es anfangs noch ein geringes Detail-Level wählt und diese im Verlauf anhebt [vgl. SM02, S. 382].

Die Effektivität von KERMIT wurde in einem Versuch mit 62 freiwilligen Studierenden untersucht [siehe SM02, S. 382]. Mittels einer Kontrollgruppe, die lediglich eine reduzierte Version von KERMIT verwenden durfte, wurden die Noten der teilnehmenden Studierenden nach zwei unabhängigen Tests verglichen. Jene Studierende, die zuvor KERMIT zur Übung verwendeten, schnitten bei dem zweiten Test besser ab

als jene aus der Kontrollgruppe [vgl. SM02, Tabelle 2]. Die Autoren schlussfolgern daraus, dass KERMIT zum Lernerfolg der Studierenden beitrug [vgl. SM02, S. 385].

MonstER Park MonstER Park ist eine Webanwendung mit der Studierende spielerisch an die Erstellung von ER-Diagrammen herangeführt werden sollen. Sie unterscheidet sich dahingehend von anderen Lernmaterialien, dass Studierende das ER-Diagramm schrittweise aufbauen, ohne vorab die gesamten Anforderungen zu kennen. Die eigentliche Aufgabe wird in Form einer fiktiven Geschichte erzählt und in kleine Teilaufgaben unterteilt.

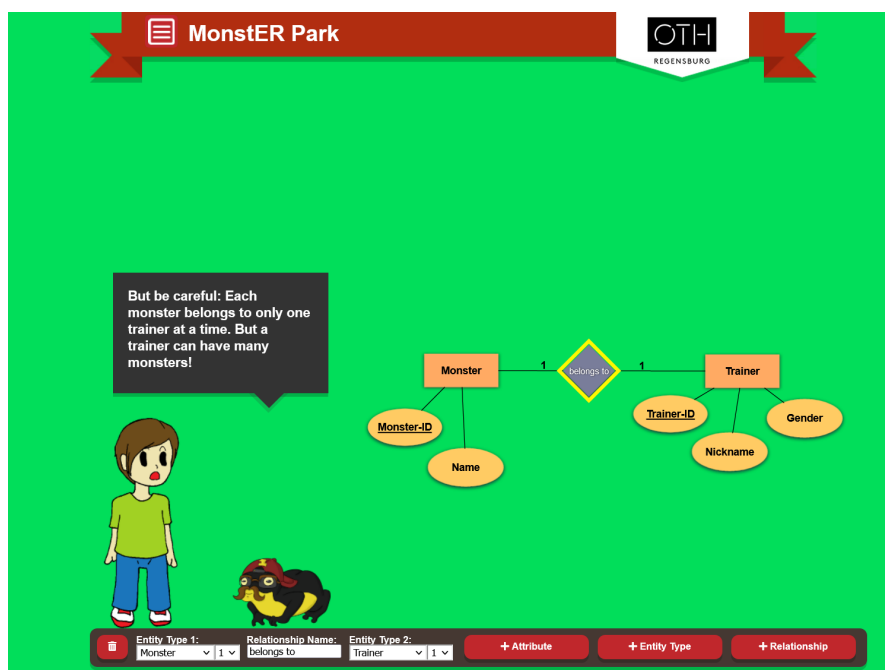


Abbildung 3.2: MonstER Park Benutzeroberfläche

Abbildung 3.2 zeigt eine Momentaufnahme von MonstER Park während der Bearbeitung einer Teilaufgabe. Auf der linken Seite sind zwei Charaktere zu sehen, die im Dialog die Geschichte und Aufgabenstellung erzählen. Auf der rechten Seite befindet sich das ER-Diagramm, welches schrittweise aufgebaut wird. Die Modellelemente sind anklickbar; Das aktuell ausgewählte Element wird farblich hervorgehoben (gelb umrandet). Im unteren Bereich der Oberfläche ist eine Leiste mit Bedienelemente zu sehen. Durch das Betätigen der Knöpfe auf der rechten Hälfte können neue Modellelemente zum Diagramm hinzugefügt werden. Sofern ein Element ausgewählt ist, in diesem Fall die Beziehung „belongs to“, werden weitere Bedienelemente zur Bearbeiten dessen in der linken Hälfte angezeigt.

Eine weitere Besonderheit der Aufgabenstellung ist, dass **MonstER Park** zu Beginn klare Anweisungen gibt, welche Elemente wie bearbeitet bzw. hinzugefügt werden sollen. Dadurch soll die Anwendung auch Studierenden ohne jegliche Vorkenntnis in der Modellierung zugänglich sein [vgl. Sch20, S. 150, 152]. Im weiteren Verlauf der Bearbeitung der Aufgaben werden diese Hilfestellungen dann reduziert.

Die Auswertung der Benutzereingaben ist binär: Bei richtiger Eingabe wird die Erzählung fortgesetzt, andernfalls müssen die Studierenden so lange das ER-Diagramm bearbeiten, bis es den antizipierten Vorgaben entspricht. Es wird kein Feedback oder Hilfestellung basierend auf den Eingaben gegeben. Allerdings werden unterschiedliche Benennungen zugelassen werden.

Die Geschichte bzw. Aufgabe muss manuell in einem JSON-Format einprogrammiert werden. Zu jeder Teilaufgabe müssen zudem die Aktionen hinterlegt werden, die zur Erfüllung dieser nötig sind. Aktuell wird nur eine Aufgabe angeboten.

Die Anwendung ist zum Zeitpunkt dieser Arbeit online ohne Zugriffsbeschränkung verfügbar. Alternativ kann sie auch offline und lokal ausgeführt werden [siehe Sch20, S. 153]. Weiter verfügt die online Version über die Möglichkeit am Ende der Bearbeitung ein Zertifikat zu erstellen. Dieses beinhaltet unter anderem einen Screenshot des erstellten ER-Diagramms, sowie ein einzigartiges Identifizierungszeichen. So können Lehrende die korrekte Aufgabenbearbeitung überprüfen.

Zur Evaluation dieser Anwendung wurde kein ausführlicher Versuch durchgeführt. Es wurden lediglich die Zugriffsstatistiken ausgewertet und auf Feedback eingegangen.

ER-Trainer Der **ER-Trainer** ist eine eigenständige Webanwendung zum Üben von ER-Diagrammen. Allerdings können keine vollständigen ER-Diagramme erstellt werden, sondern die Studierenden werden neben einer Textaufgabe mit einem unvollständigen ER-Diagramm konfrontiert. Bereits gegeben sind bis zu vier Entitäten¹ und n-näre oder rekursive Beziehungen zwischen diesen. Daneben werden auch Aufgaben zur Generalisierung und Spezialisierung aus dem EERM gestellt. Die Studierenden sollen die Kardinalitäten und Eigenschaften der Beziehungen bestimmen.

Der **ER-Trainer** dient lediglich zur Übung. Die Eingaben werden lokal im Browser überprüft und werden an keinen externen Server übermittelt¹.

Die Ansicht einer Aufgabe im **ER-Trainer** wird in Abbildung 3.3 festgehalten. Neben der Aufgabenstellung und dem unvollständigen ER-Diagramm sind weitere

¹siehe https://github.com/EILD-nrw/er_trainer/blob/225e80e693891bff2cc246cfbef0d4577af82df1/README.md, aufgerufen am 11.04.2023

ER-Trainer DE ▾

Bitte wählen Sie den zu der Phrase passenden Beziehungstyp in der Auswahlbox aus! Notation: Abrial ▾ [Legende](#)

Phrase 47/50: Der öffentliche Dienst möchte seine aktuell an Firmen vergebenen Aufträge verwalten.

Firma: Auftrag:

Die Abrial bzw. (min,max)-Notation gibt für jeden an einer Beziehung beteiligten Entitätentyp an, mit wie vielen Entitäten auf der anderen Seite eine Entität dieses Typs mindestens und höchstens in Beziehung steht.

Abbildung 3.3: Ansicht einer Aufgabe im ER-Trainer

Bedienelemente zu erkennen. Mit den Auswahlfeldern direkt unter dem Diagramm können die Studierenden die Kardinalitäten der Beziehung angeben. Automatisch wird diese auch im Diagramm selbst dargestellt. Standardmäßig in der Abrial Notation, wahlweise aber auch in Anderen.

Die Studierenden können ihre Eingaben von der Anwendung überprüfen lassen. Sollte die Eingabe falsch sein, wird ein oder mehrere Hinweise angezeigt. Die Studierenden erhalten die Chance ihre Eingabe zu korrigieren. Außerdem können sie sich bei falscher Eingabe die korrekte Lösung anzeigen lassen und sogar einzelne Aufgaben überspringen. Alle Aufgaben, ihre Lösungen und Hinweise müssen zuvor in die Anwendung einprogrammiert werden.

ER-REL-Trainer Der **ER-REL-Trainer** ähnelt dem **ER-Trainer** im Aufbau und in den Aufgaben. Allerdings sind hier bereits die Kardinalitäten und Hierarchien zwischen den Entitäten gegeben. Die Studierenden sollen das ER-Diagramm nun in ein logisches relationales Schema überführen². Dazu müssen sie die Schlüssel und Schlüsselattribute der vordefinierten Tabellen angeben. Auch hier werden die Eingaben wieder lokal überprüft.

²siehe https://github.com/EILD-nrw/er_rel_trainer/blob/464764947b7b45bc0c1d2ebfd2fb3f1ca0429232/README.md, aufgerufen am 11.04.2023

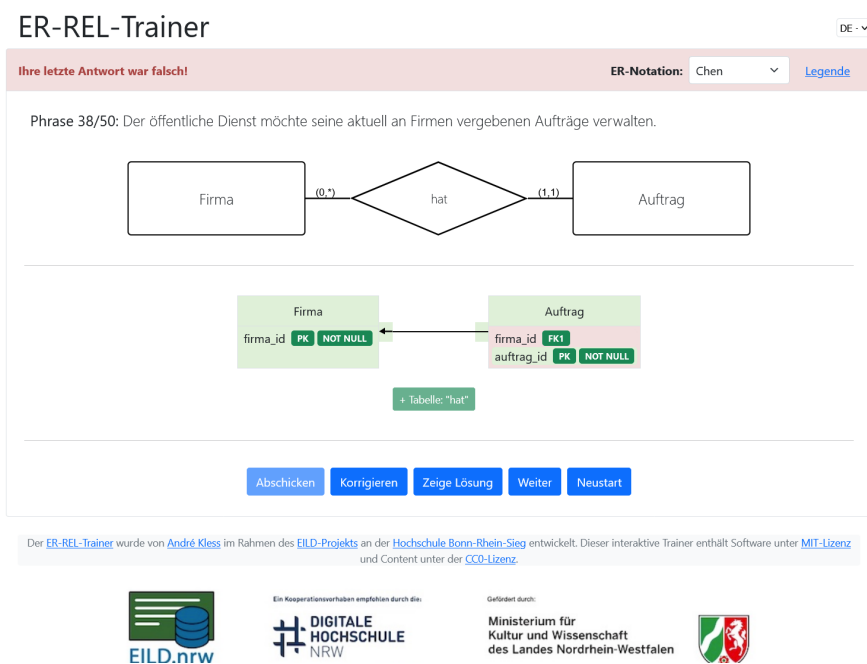


Abbildung 3.4: Überprüfung der Eingabe im ER-REL-Trainer

Abbildung 3.4 zeigt eine Momentaufnahme während der Überprüfung einer Eingabe. Die Tabellen „Verein“ und „Kontaktperson“, sowie deren Attribute und die Verbindungen (Pfeil), sind teilweise grün hinterlegt. Dadurch wird hervorgehoben welche Elemente bereits korrekt eingegeben wurden. Die rot hinterlegten Eigenschaften sind noch falsch. Die Oberfläche ist ähnlich der des ER-Trainers (vergleiche Abbildung 3.3) aufgebaut und verwendet den gleichen Ablauf von Überprüfung, Lösung und Überspringen von Aufgaben.

3.2.2 UML-Klassendiagramme

Die Unified Modeling Language (UML) wird im Bereich der Softwareentwicklung zum Beschreiben komplexer Modelle und Architekturen eingesetzt. Durch Diagramme kann so die Struktur von Anwendungen und Daten grafisch dargestellt werden. In der Objektorientierten Programmierung (OOP) werden bspw. Klassendiagramme durch UML ausgedrückt. Klassen werden mit ihren Attributen, in diesem Fall Instanzvariablen und Methoden, erfasst. Die Hierarchie und das Verhalten zwischen den Klassen werden ebenso abgebildet.

Im Kontext von Datenbanken wird primär auf das Entity-Relationship-Modell (ERM) gesetzt. ER-Diagramme werden zur Modellierung der Daten in einer Datenbankenanwendung eingesetzt, um einzelne Entitäten und deren Beziehungen zu

beschreiben. Laut Soler et, al. eignet sich das ERM lediglich für den Entwurf von Datenbanken, wohingegen UML als universelles Werkzeug eingesetzt werden kann [vgl. Sol+10, S. 973]. Aufgrund der Ähnlichkeiten beider Modellierungssprachen, sollte auch UML zum Entwurf von Datenbanken verwendet werden können, argumentieren sie weiter.

In diesem Abschnitt werden interaktive Lernmaterialien vorgestellt, die Studierende beim Erlernen von UML-Diagrammen unterstützen sollen. Der Fokus liegt auf der Anwendung im Bereich Datenbanksysteme. Das Lernziel ist, selbständig komplexe Probleme mithilfe von UML darstellen zu können.

AutoER Foss et, al. formulieren in „Automatic Generation and Marking of UML Database Design Diagrams“ [FUL22] Anforderungen an ein System zur automatischen Erstellung von Aufgaben und Auswertung der Lösungen von UML-Diagrammen zum Datenbankentwurf. Sie präsentieren ein von ihnen entwickeltes System, AutoER, welches diese Anforderungen erfüllt.

Über ein Webinterface können Studierende Aufgaben lösen und online zur Korrektur einreichen. Eine Beschreibung des Modells ist in Textform vorgegeben. Studierende sollen davon ein vollständiges UML-Diagramm ableiten. Anders als bei anderen interaktiven Materialien kann das UML-Diagramm allerdings nicht frei gezeichnet werden. Stattdessen wird es durch Interaktion mit dem Text erstellt. Wie in Abbildung 3.5 dargestellt, können Studierende Passagen im Text markieren und dann Auswählen, wie diese in das Diagramm integriert werden sollen.

Anschließend können die Studierenden ihre Lösung einreichen. Die Auswertung erfolgt automatisch und gibt Studierenden somit unmittelbar Feedback. Dazu wird das Diagramm intern in eine textliche Repräsentation konvertiert und an einen externen Server übermittelt. Auf dem Server ist eine oder mehrere Musterlösungen hinterlegt, mit denen die Eingabe dann verglichen wird. Anhand verschiedener Merkmale, wie z. B. der Namen der Entitäten, ihrer Attribute, den Beziehungen und der Wertigkeiten, usw., wird eine Bewertung vorgenommen. Die Wertigkeit der einzelnen Merkmale kann durch Lehrende im Vorfeld angepasst werden.

Dadurch, dass die Elemente im Diagramm aus dem Aufgabentext heraus erstellt werden, kann eine hohe Genauigkeit beim Vergleich zwischen Eingabe und Musterlösung erzielt werden [vgl. FUL22, S. 628]. In anderen Systemen ist die Erkennung relevanter Merkmale, durch z. B. Abweichungen in der Benennung von Elementen, schwieriger. Ein Nachteil ist allerdings, dass die Aufgabenstellung in diesem Fall möglicherweise schon zu viel über die Lösung preisgibt [FUL22, S. 628].

Eine weitere Komponente der Anwendung ist die automatische Generierung von Aufgaben. Anhand von Parametern zur Anzahl an Entitäten, Attributen und

Kapitel 3. Ergebnisse

3.2. Materialien

The screenshot displays the AutoER web interface. At the top, it shows 'Check Mark' and 'Attempts: 3/Infinity'. The main area contains a task description with several paragraphs: 'There are multiple hospitals in the medical system. A hospital is identified by its name and has a location.', 'A doctor is identified by their medical number and has a name. Each hospital has a single doctor as a manager, and a doctor may manage only one hospital.', 'Doctors are located in hospitals. A doctor may be located in more than one hospital. A doctor located at a hospital has an office number and a salary paid by that hospital.', 'A patient is identified by their health id and also has a name and gender.', 'A patient visits a doctor at a particular hospital. Each visit is identified for a particular visit by name.', and 'At a visit zero or more tests are run each with a cost and an outcome identified for a particular visit by name.'

Below the text is an ER diagram with the following entities and relationships:

- Patient**: healthId (PK), name, gender
- Test**: testName (PPK), cost, outcome
- Visit**: visitId (PK), date
- Hospital**: hospitalName (PK), location
- Doctor**: medicalNum (PK), name
- LocatedIn**: officeNum, salary

Relationships and cardinalities:

- Patient (1..1) to Visit (0..*)
- Visit (0..*) to Hospital (0..*)
- Hospital (1..1) to Doctor (1..1)
- Doctor (1..1) to LocatedIn (0..*)
- LocatedIn (0..*) to Hospital (0..1)
- Visit (1..1) to Doctor (1..1)
- Visit (0..*) to Test (0..*)

On the right, the 'Manage Entities' panel shows a list of entities with their attributes and primary keys:

- Hospital**: hospitalName {PK}, location
- Doctor**: medicalNum {PK}, name
- LocatedIn**: officeNum, salary
- Patient**: healthId {PK}, name, gender
- Visit**: visitId {PK}, date
- Test**: testName {PPK}

A context menu is open over the diagram, showing options: 'Add entity', 'Add attribute', 'Hospital', 'Doctor', 'LocatedIn', 'Patient', 'Visit', and 'Test'.

Abbildung 3.5: AutoER Webinterface: Bearbeitung einer Aufgabe [aus FUL22, S. 628]

Relationen, wird ein zufälliger Aufgabentext erzeugt. Dies hat den Vorteil, dass für Studierende individuelle Aufgaben erzeugt werden können. Ebenso lässt sich damit der Schwierigkeitsgrad anpassen.

Bei einer Umfrage, gab ein Großteil der Studierenden positives Feedback hinsichtlich der einfachen Bedienung der Anwendung an [vgl. FUL22, S. 630]. Allerdings wurden auch fehlende Features, wie etwa eine detaillierte Korrektur der Darstellung des Diagramms, angemerkt [vgl. FUL22, S. 630]. In einem von Foss et, al. durchgeführten Experiment wurde das Verhalten und die Leistung von 44 Studenten, unter Einsatz von AutoER, analysiert. Letztendlich konnte keine Aussage darüber getroffen werden, dass AutoER tatsächlich zum Lernerfolg beitrug [siehe FUL22, S. 631]. Zum einen wurde die Anwendung nicht mit anderer Software verglichen und zweitens ist die Datenmenge laut Autoren unzureichend [vgl. FUL22, S. 631].

UML-Klassendiagramme für ACME-DB Die von Soler et, al. entwickelte Applikation wurde in eine bereits bestehende Umgebung, ACME-DB, integriert. Es erweitert die dort bereits angebotenen Lernmodule um ein UML-Modul. In einer webbasierten, grafischen Oberfläche können Studierende ein UML-Klassendiagramm zeichnen. Von einer Aufgabenstellung müssen sie die Anforderungen ableiten. Abbildung 3.6 zeigt einen Screenshot des Moduls.

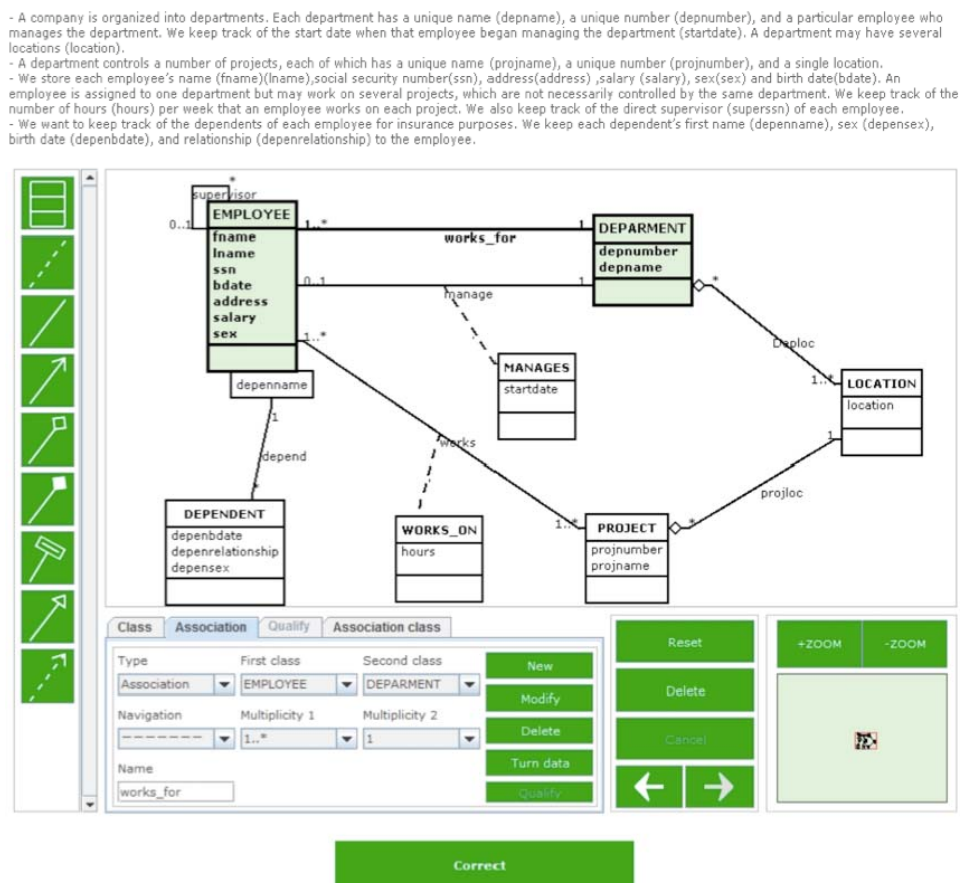


Abbildung 3.6: Interface zum Erstellen eines UML-Klassendiagramms mit Aufgabenstellung in ACME-DB [aus Sol+10, Abbildung 6.]

In den zuvor definierten Anforderungen halten Soler et, al. fest, dass die Applikation eine automatisierte Korrektur der Eingaben durchführen soll [vgl. Sol+10, S. 975]. Auch wird wieder die Eingabe mit einer oder mehreren Musterlösungen auf einem externen Server verglichen. Entsprechend der Unterschiede werden Nachrichten generiert, die die Studierenden an die Musterlösung heranzuführen sollen [vgl. Sol+10, S. 977]. Gleichzeitig erhalten Lehrende dadurch eine Übersicht über häufige Fehler in den Abgaben.

Bei der Erstellung der Aufgaben müssen die Klassen und ihre Attribute in einer Domain-specific Language (DSL) definiert werden. Das daraus resultierende Schema wird dann als eine Musterlösung verwendet. Der Abgleich ist so implementiert, dass die Klassen anhand ihrer Attribute in der Musterlösung erkannt werden. Soler et al. gehen jedoch nicht näher darauf ein, wie genau die Attribute und Klassen zugeordnet werden. Sobald eine Klasse aus der Eingabe zugeordnet werden konnte, werden weitere Eigenschaften, wie z. B. die Beziehungen zu anderen Klassen, ausgewertet. Die Benennung der Klasse wird nicht berücksichtigt [siehe Sol+10, S. 977].

In einem Experiment zur Evaluation teilten die Autoren insgesamt 48 Studenten in zwei Gruppen auf. Die eine Gruppe verwendete die vorgestellte Anwendung, die andere diente als Kontrollgruppe. Dann wurde den Studenten mehrere Aufgaben gestellt, zu denen sie ein UML-Diagramm anfertigen sollten. Die Aufgaben dienten lediglich als Übung. Dennoch ist zu erkennen, dass jene Studierende, die die Anwendung einsetzten, mehr Aufgaben erfolgreich bearbeitet haben. Diese Tendenz konnte auch in einer im Anschluss daran stattfindenden einheitlichen Prüfung bestätigt werden. Die Autoren merken jedoch an, dass die Ergebnisse nur experimentell seien und zukünftig mit einer größeren Anzahl an Studierenden durchgeführten werden müssten [vgl. Sol+10, S. 978].

Vor allem lobten Studierende und Lehrkräfte die einfache Bedienung und Zugänglichkeit des Systems [vgl. Sol+10, S. 978]. Die Lehrkräfte gaben ebenfalls an, dass der Lernfortschritt der Studierenden besser nachverfolgt werden kann und somit gezielter auf einzelne Studierende eingegangen werden kann [vgl. Sol+10, S. 978].

3.2.3 Normalisierung

Unter Normalisierung wird im Allgemeinen der Prozess verstanden, ein logisches, relationales Datenbankschema hinsichtlich der Qualität der Datenhaltung zu optimieren. Das Schema wurde zuvor aus einem konzeptionellen Datenbankentwurf (z. B. in Form eines ERM) abgeleitet. Weil der konzeptionelle Entwurf keine direkten Anforderungen an das Datenmodell stellt, können bei der Überführung u. a. Redundanzen entstehen. Anhand von fest definierten Regeln kann das logische Schema schrittweise in sog. Normalformen überführt werden. Diese Normalformen erfüllen spezielle Eigenschaften, die die Integrität der Daten gewährleisten sollen. Ein Schema gilt als „normalisiert“ sobald es die Anforderungen der dritten Normalform (3NF) erfüllt [vgl. Kud15, S. 80].

NORMIT Eine der ersten Anwendungen zum Üben von Normalisierungen, NORMIT, wurde im Jahr 2002 an der „University of Canterbury“ entwickelt. Die Anwendung kann ergänzend zur Lehre verwendet werden, weil keine Konzepte vermittelt werden [vgl. Mit03b, S. 566 f.].

Eine Besonderheit liegt bei der Lernmethode, die hier zum Einsatz kommt: Im Anschluss an die Bearbeitung der praktischen Aufgaben müssen Studierende ihre Eingaben über Multiple-Choice Fragen begründen. Dadurch sollen die Regeln zur Normalisierung vertieft werden und einfaches Raten verhindert werden. In der Fachliteratur wird diese Lernmethode auch „Rechtfertigung“ (engl. „self-explanation“) genannt [siehe Mit03b, S. 565].

Der Normalisierungsprozess umfasst mehrere Regeln und Zustände (z. B. Normalformen), die schrittweise auf ein relationales Schema angewendet werden müssen. Dementsprechend ist auch diese Anwendung strukturiert. Die Studierenden müssen in Aufgaben das Schema schrittweise bearbeiten. Allerdings bestehen diese Aufgaben lediglich aus Texteingaben und Multiple-Choice Fragen.

Zur Evaluation der Eingaben wird CBM (siehe auch Abschnitt 3.2.1) eingesetzt. Anhand von 54 Bedingungen (engl. „constraints“) wird der Zustand nach jeder Aufgabe überprüft. Wenn eine Bedingung nicht erfüllt ist, also z. B. eine Regel nicht eingehalten wurde, wird eine Fehlermeldung generiert. Anfangs merkt diese nur an, dass etwas falsch ist. Nach weiteren Einreichungen werden die Fehlermeldungen detaillierter und geben letztendlich Hinweise zur Richtigstellung.

Normalization Educational Tool In [SK16] stellen Stefanidis und Koloniari eine Anwendung zum Normalisieren von relationalen Datenbanken vor. Basierend auf einem eingegebenen Schema kann die Anwendung automatisch Relationen ableiten, die die Boyce-Codd Normalform (BCNF) erfüllen. Das Hauptmerkmal der Anwendung ist die Darstellung der schrittweisen Normalisierung. Durch dieses Feature sei die Anwendung sowohl in der Lehre als auch zum Lernen einsetzbar [vgl. SK16, S. 1 ff.]. Anhand von verschiedenen Beispielen können Lehrende bspw. den Prozess zur Überführung in die BCNF besser veranschaulichen. Studierende können die Anwendung zum Üben und Validieren ihrer Lösungen einsetzen [siehe SK16, S. 1 ff.].

Abbildung 3.7 zeigt zwei Ausschnitte aus dem Normalization Educational Tool. In Abbildung 3.7a ist das Interface zur Definition des relationalen Datenbankschemas zu sehen. Neben den Attributen und ihren Datentypen der Relation müssen auch die funktionalen Abhängigkeiten (rechts) angegeben werden. Das Schema kann jederzeit in einer Datei abgespeichert oder von einer Datei geladen werden. Über einen Knopf in der Toolbar (im oberen Bereich) kann das Fenster zum schrittweisen Normalisieren, zu sehen in Abbildung 3.7b, geöffnet werden. Dort müssen die Anwender zuerst eine Relation und eine funktionale Abhängigkeit (auf der linken Seite) auswählen, die abgeleitet werden soll. Auf der rechten Seite wird die Ableitung, nach Betätigen des Knopfes am unteren Rand, detailliert beschrieben. Abhängig vom Vorgang entstehen weitere Relationen, die genauso dargestellt werden.

Kapitel 3. Ergebnisse

3.2. Materialien

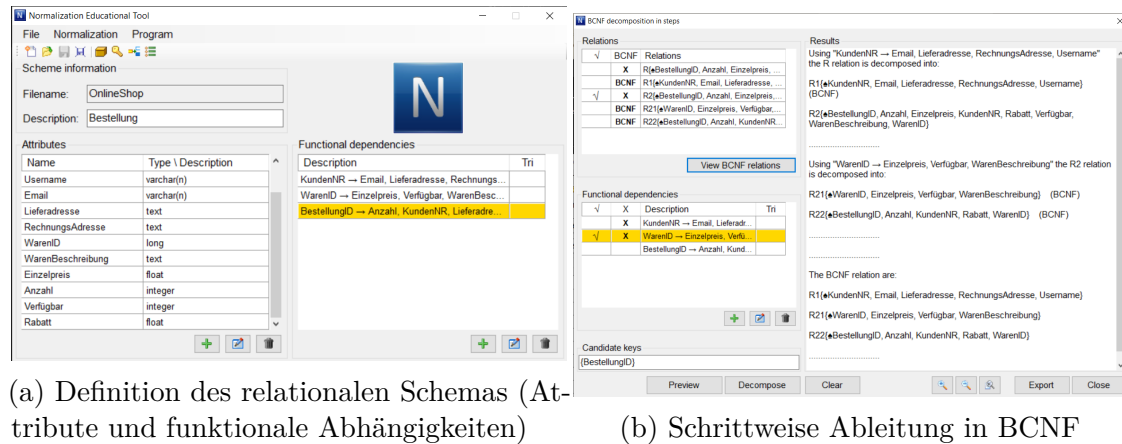


Abbildung 3.7: Verschiedene Interfaces des Normalization Educational Tool

NF-Trainer Der Normalformen-Trainer (NF-Trainer) ist eine Webanwendung, die als Teil des EILD.NRW Projektes (siehe Abschnitt 2.1) frei verfügbar ist. Er dient als Lernumgebung zum Üben von Normalisierungen. In aktuell insgesamt 8 Aufgaben sollen Studierende ein vorgegebenes Schema schrittweise in die drei Normalformen überführen. Eine Aufgabe besteht aus einer kurzen Einleitung und einer Tabelle mit Beispieldaten. In unterschiedlichen Fragen müssen Studierende abwechselnd die aktuellen Relationen beurteilen, funktionale Abhängigkeiten erkennen und Schlüsselattribute bestimmen. Bei falschen Angaben kann die Musterlösung eingesehen werden.

In Abbildung 3.8 ist ein Ausschnitt einer Aufgabe aus dem NF-Trainer zu sehen. In der oberen Hälfte ist der einleitende Satz und die Tabelle mit den Beispieldaten zu sehen. Darunter befinden sich die Eingabemasken für die funktionalen Abhängigkeiten. Es müssen immer 4 Abhängigkeiten angegeben werden. Je Abhängigkeit können ein oder mehrere Primärschlüsselattribute und ihre Nichtschlüsselattribute ausgewählt werden. Über den Knopf oben rechts, kann eine generelle Definition und Anleitung zur Überführung in die Normalformen abgerufen werden.

3.2.4 Datenstrukturen

Datenstrukturen beschreiben, wie Informationen (Daten) im Speicher hinterlegt werden. Unterschiedliche Datenstrukturen werden eingesetzt, um verschiedenen Anforderungen hinsichtlich ihres Platzbedarfs und der Zugriffsgeschwindigkeit gerecht zu werden. Bei der Umsetzung in Software spielen sie daher eine wichtige Rolle und können, bei richtiger Verwendung, den Programmablauf signifikant beschleunigen oder den Speicherverbrauch minimieren.

NF-Trainer



Skriptverkauf - Funktionale Abhängigkeiten

Definitionen und Hinweise

In der folgenden Tabelle wird beschrieben, welche Personen in ihren Veranstaltungen Skripte in welcher Anzahl verkaufen.

Veranstaltungsnummer	Veranstaltungsname	Personalnummer	Name	Skriptnummer	Preis	Anzahl
112	Englisch	198	Schulz	2	25	4
112	Englisch	237	Lange	9	44	5
112	Englisch	11	Meyer	2	25	4
202	Spanisch	198	Schulz	4	22	9

Bestimmen Sie alle funktionalen Abhängigkeiten, indem Sie auf die entsprechenden Spalten (Primärschlüssel und abhängige Spalten) klicken.

Primärschlüssel

Veranstaltungsnummer

Veranstaltungsname

Personalnummer

Name

Skriptnummer

Preis

Anzahl

Spalten

Veranstaltungsnummer

Veranstaltungsname

Personalnummer

Name

Skriptnummer

Preis

Anzahl

Primärschlüssel

Veranstaltungsnummer

Veranstaltungsname

Personalnummer

Name

Skriptnummer

Preis

Anzahl

Spalten

Veranstaltungsnummer

Veranstaltungsname

Personalnummer

Name

Skriptnummer

Preis

Anzahl

Abbildung 3.8: Bearbeitung einer Aufgabe im NF-Trainer

Auch DBMS verwenden intern unterschiedliche Datenstrukturen und Algorithmen, um diverse Probleme möglichst effizient zu lösen. So kommen bspw. Hash-Tabellen, verschiedene Baumstrukturen, Listen usw. zum Einsatz. Die Anwendung ist aber keineswegs auf DBMS beschränkt. Daher werden Datenstrukturen und Algorithmen, unabhängig von Datenbanksystemen, in den meisten Informatik-Studiengängen gelehrt.

In diesem Abschnitt werden Lehrmaterialien vorgestellt, die Datenstrukturen und Algorithmen visuell veranschaulichen sollen.

VisuAlgo Auf visualgo.net können Datenstrukturen und Algorithmen visuell erkundet werden. Die Website umfasst aktuell verschiedene Arten von Listen, Sortieralgorithmen, Baumstrukturen, Graphen und vieles mehr. Studierende können durch Benutzereingaben die Daten verändern und dabei das Verhalten der Datenstruktur beobachten. Die Teilschritte des Algorithmus' zur Überführung zum neuen Zustand werden animiert. Halim et, al. [Hal+12] entwickelten 2012 das Projekt, um Studierenden die Datenstrukturen besser erklären zu können. Seitdem wurden weitere Inhalte hinzugefügt und die Plattform weiterentwickelt [siehe Hal+12, S. 67;

Eri14, S. iii]. Inzwischen ist auch ein Quiz eingebaut, mit dessen Hilfe Studierende ihr Wissen testen können.

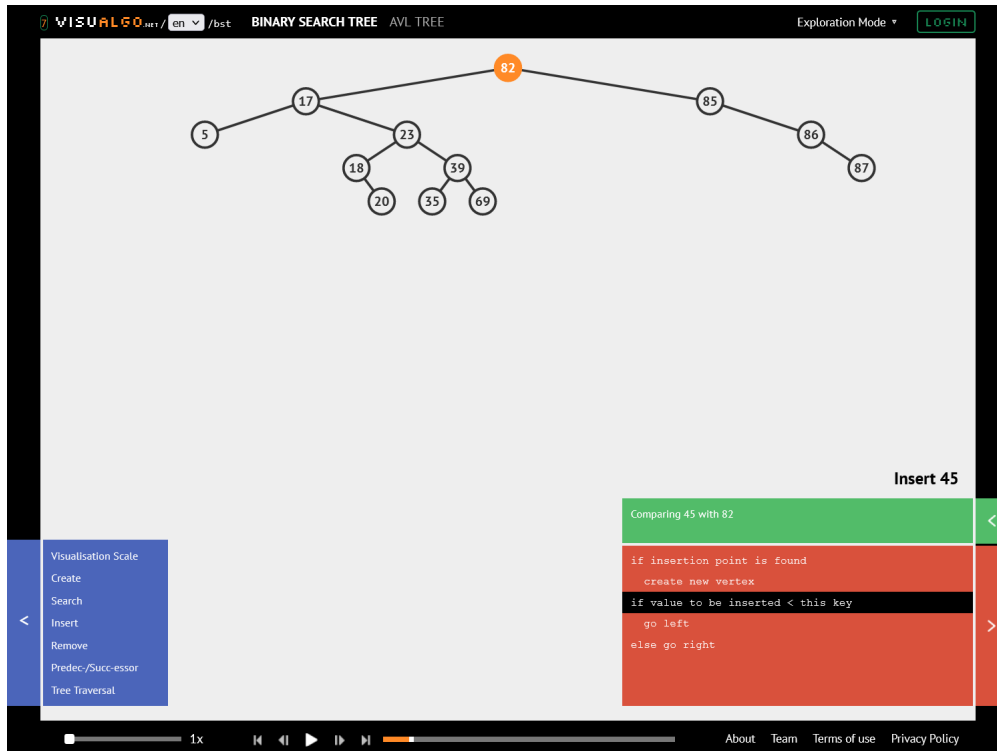


Abbildung 3.9: Darstellung eines Binärbaums in VisuAlgo

Für jede Datenstruktur gibt es eine separate Seite, auf der unterschiedliche Aktionen ausgeführt werden können. Abbildung 3.9 zeigt die Darstellung eines Binärbaums. Die Baumstruktur wird im Mittelpunkt des Bildschirms entsprechend präsentiert. Im unteren linken Bereich befindet sich ein Bedienelement zum Steuern des Baums. Die grundlegenden Funktionen, wie Erstellen, Einfügen, Suchen und Löschen, können hierdrüber ausgeführt werden. Auf der rechten Seite wird Pseudocode angezeigt, welcher den Algorithmus der jeweiligen Aktion beschreibt. In diesem Fall schwarz hinterlegt, ist die Zeile, die aktuell ausgeführt und animiert wird. Darüber, grün hinterlegt, wird eine genauere Beschreibung des aktuellen Teilschritts angezeigt. Über die Bedienelemente am unteren Rand kann die Animation gesteuert werden. Die Darstellung des Baums verändert sich entsprechend. Wird ein Knoten ausgewählt so ist dieser farblich markiert (orange). Wird ein neuer Knoten eingefügt, so ist dies ebenfalls zu erkennen.

B-Baum-Animation Die B-Baum-Animation basiert auf dem B-Baum-Zeichner. Beide Webanwendungen werden im Rahmen des EILD.NRW Projektes angeboten.

Sowohl das Einfügen in einen, als auch das Löschen aus einem B-Baum, werden animiert. Dazu wird der Algorithmus in Teilschritte zerlegt, welche einzeln beschrieben und visuell dargestellt werden. Die Animation wird durch Benutzereingaben der Studierende gestartet. Die **B-Baum-Animation** stellt allerdings nur einfache B-Bäume dar und keine weiteren Datenstrukturen. Außerdem kann nicht im Baum gesucht werden.

3.2.5 Structured Query Language

SQL steht für Structured Query Language und ist ein Sprachkonstrukt, um mit Relational Database Management Systems (RDBMS) zu interagieren. In dem Sprachkonstrukt sind Anweisungen enthalten, um Daten strukturiert zu speichern, abzufragen, zu verändern und zu löschen.

Mit dem Ziel eine herstellerunabhängige Sprache für Datenbanken zu entwickeln, wurde von der „International Organization for Standardization“ (ISO) bereits im Jahr 1987 SQL standardisiert. Seitdem wird der Standard kontinuierlich erweitert und angepasst. Die aktuelle Version wurde im Jahr 2016 veröffentlicht und die nächste ist bereits in Konzeption [siehe Int16].

In vielen RDBMS Produkten hat sich SQL seit der Standardisierung als Sprache etabliert. Allerdings kommt es vor, dass die Sprache in den verschiedenen Hersteller Implementierungen vom Standard abweichen, um die Funktionalität zu modifizieren oder zu erweitern. Diese Abweichung nennt sich SQL-Dialekt.

Die Sprache SQL kann in vier Kategorien unterteilt werden [vgl. Kud15, S. 101]:

- Mit der Data Definition Language (DDL) können Datenstrukturen definiert, verändert und gelöscht werden.
- In der Data Manipulation Language (DML) sind Sprachkonstrukte enthalten, um Datensätze zu erstellen, zu verändern und zu löschen.
- Die DQL wird genutzt um Datensätze strukturiert abzufragen.
- Ein etwas seltener verwendeter Teil der SQL ist die Data Control Language (DCL), welche dafür zuständig ist Berechtigung zu vergeben und zu entziehen.

Wie bereits in der quantitativen Auswertung der recherchierten interaktiven Lehr- und Lernmaterialien ersichtlich war, ist der Themenbereich SQL der am häufigsten vorkommende (vgl. Abbildung 2.4). Im Folgenden wird genauer auf die einzelnen SQL-Materialien und die Literatur eingegangen.

Kapitel 3. Ergebnisse

3.2. Materialien

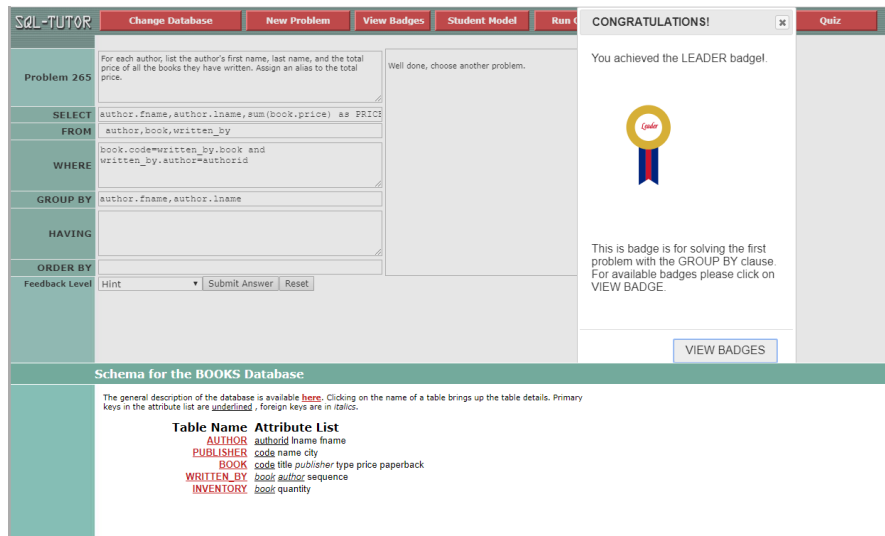


Abbildung 3.10: SQL-Tutor – Webinterface nach der Integration von Gamification Elementen [TMS20]

SQL-Tutor Als eines der ersten interaktiven Lern- und Lehrmaterialien für SQL, das in Veröffentlichungen Erwähnung fand, hat der **SQL-Tutor** eine historische Bedeutung. Der **SQL-Tutor** wurde im Jahr 1998 von Mitrovic als Desktopanwendung [Mit98] entwickelt und im Folgejahr auf eine webbasierte Version [Mit03a] portiert. Die Webversion trägt den Namen **SQLT-Web**. In den anschließenden Veröffentlichungen wird das Material weiterhin als **SQL-Tutor** bezeichneten, obwohl es sich um die Webversion handelt.

Sowohl die Desktopanwendung als auch die Webvariante soll den Studierenden Inhalte zu DQL vermitteln. Dabei geht das Material, wie folgt vor: In dem System sind viele verschiedene Aufgabenstellungen hinterlegt. Die Aufgaben bestehen aus einer textuellen Beschreibung eines Problems. Das Problem ist dabei so gestellt, dass dieses mit einer „SELECT“ Abfrage gelöst werden kann. Nachdem die Studierenden ihre Lösung entwickelt und abgeschickt haben, wird eine syntaktische und semantische Prüfung durchgeführt. Diese Prüfung fußt auf dem Vergleich zwischen Musterlösung und Eingabe der Studierenden. Mit dem Ergebnis der Prüfung wird entsprechendes Feedback generiert. Basierend auf dem Lernfortschritt wird den Studierenden mit einem fünfstufigen Feedbacksystem individuelle Hilfestellungen angeboten: Richtig/Falsch, Markierung der Fehler, Hinweise, Teil- oder Komplettlösung. Die Einstufung in die Kategorien (Anfänger, Fortgeschritten oder Experte) nehmen die Studierenden selbst vor.

Die Veröffentlichung von Mitrovic [Mit03a] beschreibt die Portierung auf die Webversion, einschließlich der Architektur, Entwicklung, Funktionsweise und Evaluation.

Besonders interessant ist die Evaluation des **SQLT-Web** an der „University of Canterbury“, welche den Grundstein für weitere interaktive SQL-Lern- und Lehrmaterialien legte. Die Evaluation setzt sich aus drei Studien zusammen, wobei zwei Studien auf der subjektiven Bewertung der Studierenden basieren und eine dritte Studie ein Experiment mit Kontroll- und Versuchsgruppe darstellt. In den ersten beiden Studien wurden die Studierende nach einer Übungseinheit mit dem **SQLT-Web** befragt. Mit dem Ergebnis, dass die befragten Studierenden die Übungseinheit als ein Lernerfolg eingestuft haben. Außerdem würde eine überwältigende Mehrheit ihren Kommilitonen eine Nutzung empfehlen. Im Gegensatz dazu wurde in der dritten Studie vor Beginn und nach einer siebenwöchigen Verwendung der Anwendung ein Wissenstest durchgeführt. Hierbei wurde eine signifikante Verbesserung der Ergebnisse der Versuchsgruppe festgestellt. Jedoch haben die Autoren, begründet durch die niedrige Anzahl an Teilnehmern (33, 34, 70) und der unkontrollierten Auswahl an Teilnehmern, auf eine Anzweiflung der Ergebnisse hingewiesen [vgl. Mit03a, S. 185 ff.].

Auch werden an dem **SQL-Tutor** fortführend neue Lerntheorien ausprobiert. Eine Weiterentwicklung zeigt z. B. die Abbildung 3.10. Hier haben die Autoren Tahir et, al. spielerische Elemente für den SQL-Tutor entwickelt und anschließend diese evaluiert. In der Evaluation ist keine signifikante Verbesserung der Lernergebnisse festgestellt worden, lediglich eine Tendenz [vgl. TMS20, S. 421 ff.].

SQL-Trainer EILD.nrw Während der Materialrecherche sind vielfach Projekte mit dem selben Namen **SQL-Trainer** vorgekommen und aufgenommen worden (siehe Tabelle 2.3). In diesem Abschnitt wird der **SQL-Trainer** aus dem Forschungsprojekt *EILD.nrw* beschrieben.

Der **SQL-Trainer** ist ein webbasiertes Material mit dem Studierende selbstverantwortlich ihre SQL-Kenntnisse prüfen und verbessern können. Im Gegensatz zu dem **SQL-Tutor** von Mitrovic kann der **SQL-Trainer** aus dem *EILD.nrw* Projekt nicht nur DQL sondern auch DML und DDL-Aufgaben stellen. Auf der Startseite wirbt der **SQL-Trainer** mit 200 DQL, 160 DML und 40 DDL-Aufgaben³.

Die Aufgaben wurden in drei Schwierigkeitsgrade (Leicht, Mittel und Schwer) unterteilt und nach Aufgabetypen (DQL, DML und DDL) kategorisiert. Zudem wurden die Aufgaben auf der Grundlage von fünf Themenbereichen (Busfahrten, Fahrradhersteller, Theater, Reisen und Fußball) entwickelt. Dabei entspricht jeder Themenbereich einem Datenbankschema. Mithilfe dieser aus dem Alltag bekannten und wiederkehrenden Themenbereiche können die Studierenden die

³siehe Startseite des SQL-Trainers <https://eild-nrw.github.io/sql-trainer/>, aufgerufen am 11.04.2023

SQL-Trainer



DQL-Trainer

Welche Angestellten verdienen zwischen 50000 und 80000 DM im Jahr? Geben Sie Ang_nr, Nachnamen und Vornamen aus!

Editor

```
1 SELECT Ang_nr, Nachname, Vorname
2 FROM Angestellte
3 WHERE Gehalt*12 > 50000 AND Gehalt*12 < 80000;
```

Ausführen **Lösung anzeigen** **Neue Aufgabe**

Tabellen

- ABTEILUNGEN
- ANGESTELLTE
- ARTIKEL
- AUFTRAEGE
- AUFTRAGSPOSITIONEN
- GEH_KLASSEN
- KUNDEN
- LAGER

▼ Ausgabe - Richtig!

ANG_NR	NACHNAME	VORNAME
4	Schmidt	Otto
5	Weber	Anna
6	Frisch	Paul
7	Frisch	Paula

Abbildung 3.11: SQL-Trainer EILD.nrw – DQL-Aufgabe

Schemas schneller verstehen. Anhand der drei vorher erwähnten Kriterien (Schwierigkeitsgrad, Themenbereich und Aufgabentyp) selektiert der **SQL-Trainer** aus seinem Repertoire eine zufällige Aufgabe. Die Kriterien wählen die Studierenden vor Trainingsbeginn selbst aus.

Nach erfolgreicher Auswahl der Kriterien durch die Studierenden präsentiert der **SQL-Trainer** den Trainingsbereich mit einer Aufgabe. Dabei ist der Trainingsbereich in vier Sektionen unterteilt (siehe Abbildung 3.11). In der ersten Sektion befindet sich die Aufgabenstellung in Textform. Danach folgt ein „Editor“ Bereich, indem Studierende ihre SQL-Lösungen zu der Aufgabe entwickeln können. Die darauffolgende Sektion nennt sich „Tabelle“. Hier können die Studierenden das Datenbankschema des Themenbereiches und den jeweiligen Inhalt der Tabellen erkunden. In der abschließenden Sektion „Ausgabe“ wird das Ergebnis des SQL-Befehls, welcher in dem „Editor“ entwickelt worden ist, in tabellarischer Form ausgegeben. Zusätzlich wird in dieser Sektion die Lösung der Studierenden mit der Musterlösung verglichen und das Ergebnis in binärer Form (Richtig/Falsch) ausgegeben. Falls in dem SQL-Befehl ein Syntaxfehler enthalten ist, wird an dieser Stelle die Fehlermeldung des DBMS angezeigt. Sollten die Studierenden keine Idee mehr haben, wie sie die Aufgabe lösen können, bietet der **SQL-Trainer** die Option, die vordefinierte Musterlösung anzuzeigen.

Die Ähnlichkeit zu dem **SQL-Trainer**⁴ des E-Learning Datenbank Portals „edb“ der Technischen Hochschule Köln ist strukturell und visuell deutlich erkennbar. Daher ist es nicht verwunderlich, dass der Entwickler des **EILD.nrw SQL-Trainers** Alexander Kosmehl diesen als eine Weiterentwicklung des im „edb“ Portal verfügbaren **SQL-Trainers** beschreibt⁵. Obwohl die Materialien visuelle Ähnlichkeiten aufweisen, unterscheiden sich die beiden Materialien hinsichtlich ihrer Umsetzung und Funktionalität grundlegend voneinander.

Der erste wesentliche Unterschied zeigt sich bereits bei der Auswahl des Aufgabentyps. In der edb-Version sind ausschließlich Aufgaben zu DQL verfügbar, während die DDL- und DML-Aufgaben als „demnächst Verfügbar“ deklariert sind. Weitergehend wurde das Feedbacksystem im edb-Projekt verändert, so dass die Studierenden bei semantischen Fehlern eine bessere Unterstützung erhalten. Zum Beispiel wenn bei einer Aufgabe die Filterbedingung vergessen wurden, bekommen die Studierenden den Hinweis „Da fehlt eine WHERE-Klausel!“. Diese Erweiterung des Feedbacksystem wurde von den Autoren Faeskorn-Woyke et, al. auf der DELFI 2020 vorgestellt. Insbesondere wurde auf die Entwicklung des Systems mithilfe eines Entscheidungsbaumes eingegangen [siehe FWBS20]. Durch die Analyse von gespeicherten Fehlern, die von Studierenden im **SQL-Trainer** gemacht wurden, wurde der Entscheidungsbaum entwickelt. Auch die architektonische Umsetzung unterscheidet sich. In dem **SQL-Trainer** des edb-Portals wird im Hintergrund eine Oracle DB eingesetzt, hingegen setzt das *EILD.nrw* auf eine im Browser laufende *SQLite* Datenbank. Daher ist der *EILD.nrw SQL-Trainer* eine reine Clientseitige Anwendung.

SQLBolt Mit dem webbasierten **SQLBolt**⁶ können Studierende SQL lernen und üben. Im Gegensatz zu den beiden vorher vorgestellten SQL-Materialien kann dieses Material auch ohne Vorwissen genutzt werden. Daher ist keine Vorlesung, Lehrbuch oder ähnliches nötig, um mit diesem Material SQL zu lernen. Dies stellt der **SQLBolt** mit einer vorangestellten umfangreichen Erklärung sicher.

Das Material ist in insgesamt 18 Lernabschnitte aufgeteilt, welche strukturell aufeinander aufbauen. In den Lernabschnitten werden Inhalte zu den drei Themenbereichen DQL, DML und DDL vermittelt.

Jeder Lernabschnitt besteht aus einem Erklärungsteil sowie einem Übungsteil (siehe Abbildung 3.12). Der Erklärungsteil jedes Lernabschnitts vermittelt den Inhalt durch natürlichsprachige Texte, die durch unterstützende Beispiele ergänzt werden.

⁴<https://edb2.gm.fh-koeln.de/apps/sqlmenu/>

⁵siehe README des **SQL-Trainers** <https://github.com/EILD-nrw/sql-trainer/blob/main/README.md>, aufgerufen am 11.04.2023

⁶erreichbar unter <https://sqlbolt.com>

SQLBolt
Learn SQL with simple, interactive exercises.

Interactive Tutorial

SQL Lesson 1: SELECT queries 101

To retrieve data from a SQL database, we need to write **SELECT** statements, which are often colloquially referred to as *queries*. A query in itself is just a statement which declares what data we are looking for, where to get it from, and how to transform it before it is returned. In this lesson, we are going to learn in the following exercises how to write these queries.

As we mentioned in the introduction, you can query a table (ie. Dogs), and each row in that table represents a single entity (ie. a beagle, a different colored pug, etc.). This represents the common properties shared by all entities in that table (length of tail, etc.).

And given a table of data, the basic SQL query is to select for a couple columns (properties) of interest.

Select query for a specific column:

```
SELECT column, another_column, ... FROM mytable;
```

The result of this query will be a two-dimensional copy of the table, but only with the columns specified in the query.

If we want to retrieve absolutely all the columns from a table, we can use the asterisk (*) shorthand in place of listing all the columns.

Exercise

We will be using a database with data about some of Pixar's classic movies for most of our exercises. This first exercise will only involve the **Movies** table, and the default query below currently shows all the properties of each movie. To continue onto the next lesson, alter the query to find the exact information we need for each task.

Table: Movies

Title	Director
Toy Story	John Lasseter
A Bug's Life	John Lasseter
Toy Story 2	John Lasseter
Monsters, Inc.	Pete Docter
Finding Nemo	Andrew Stanton
The Incredibles	Brad Bird
Cars	John Lasseter
Ratatouille	Brad Bird
WALL-E	Andrew Stanton
Up	Pete Docter

Exercise 1 — Tasks

1. Find the title of each film. ✓
2. Find the director of each film. ✓
3. Find the title and director of each film. ✓
4. Find the title and year of each film.
5. Find all the information about each film.

Stuck? Read this task's [Solution](#).
Solve all tasks to continue to the next lesson.

```
SELECT title, director FROM movies;
```

Abbildung 3.12: SQLBolt – Erklärungs- und Übungsteil der „SQL Lesson 1“

Im Anschluss können die Studierenden im Übungsteil ihre theoretisch erworbenen Fähigkeiten in kleinen Aufgaben üben und überprüfen. Die interaktiven Aufgaben bestehen aus einem Bereich, in dem die Aufgabe beschrieben ist, einem weiteren Bereich, in dem das Ergebnis des aktuellen SQL-Befehls in tabellarischer Form angezeigt wird, und einem Bereich, in dem der SQL-Befehl zur Lösung der Aufgabe erstellt werden soll (vgl. rechte Abbildung 3.12). Sobald der korrekte SQL-Befehl eingegeben wurde, wird die Aufgabe automatisch als abgeschlossen markiert und die nächste Aufgabe wird freigeschaltet. Weitere Hilfestellungen oder einen einstellbaren Schwierigkeitsgrad stellt das Material nicht zur Verfügung. Falls die Studierenden einmal nicht die passende Lösung erstellen können, gibt es noch die Option eine Musterlösung anzuzeigen.

SQLBolt erfordert nicht, dass die Übungen gelöst werden müssen. Daher können die Studierenden zu den nächsten Lernabschnitten springen, falls sie bereits das Wissen erlangt, oder kein Interesse haben die Inhalte zu üben.

Datenbank-Kreuzworträtsel Mit dem Datenbank-Kreuzworträtsel können die Studierenden ihr Wissen zu SQL-Themen spielerisch prüfen. Insbesondere wird in diesem Material der Fokus auf die Terminologie gelegt. Wie bereits aus dem Title ersichtlich ist, verwendet das Material ein Kreuzworträtsel, um die Termini abzufragen. Der Vorteil bei dieser Art von Rätsel besteht darin, dass die Funktionsweise allgemein bekannt ist, da diese häufig in Tageszeitungen enthalten

sind. Daher bedarf es keiner Erläuterung, um mit diesem Rätsel zu interagieren.

Das **Datenbank-Kreuzworträtsel** bietet den Studierenden zwei Einstellungsmöglichkeiten, um das Rätsel an ihren Bedarf anzupassen. Zum ersten kann das Thema ausgewählt und zum zweiten kann die Größe des Kreuzworträtsels bestimmt werden. Zum Zeitpunkt der Recherche sind zwei Themenbereiche (*SQL* und *PL/SQL*) und drei Größen (*Klein*, *Mittel* und *Groß*) in dem Material hinterlegt.

Sobald die Studierenden ihre Lösung mit der Musterlösung vergleichen möchten, wird eine Überprüfung der Eingabe durchgeführt. Die fehlerhaften Buchstaben werden rot markiert und die korrekten grün hinterlegt. Außerdem wird der erwartete Terminus neben dem Rätsel angezeigt (siehe Abbildung 3.13).

Das Tool wurde im Rahmen des „EILD.nrw“ Projekts entwickelt. Dabei fußt die Entwicklung auf einem Tool, welches bereits an der TH-Köln konzipiert und umgesetzt wurde⁷.

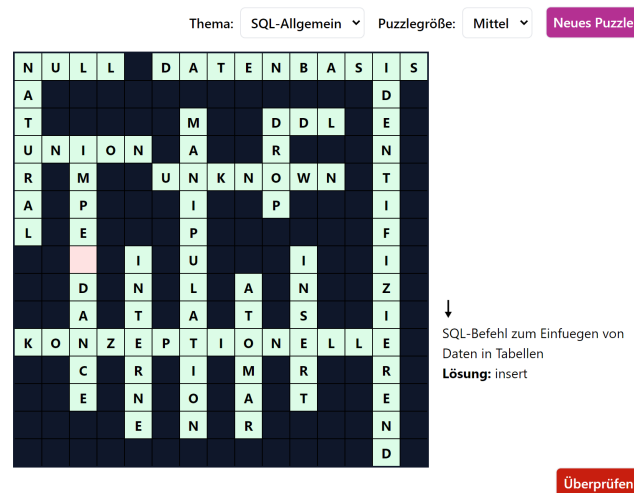


Abbildung 3.13: Ein DB-Kreuzworträtsel zu den SQL-Grundlagen

Relational Database freeCodeCamp Eine Onlineplattform, welche eine Vielzahl von Bildungsinhalte öffentlich anbietet und dabei ein breites Spektrum bietet, nennt sich Massive Open Online Courses (MOOCs). MOOCs können aufgrund ihrer einfachen Zugänglichkeit über das Internet eine große Anzahl an Interessierten Personen erreichen. Ein MOOCs ist *freeCodeCamp*.

⁷siehe README des DB-Kreuzworträtsel <https://github.com/EILD-nrw/crossword/blob/main/README.md>, aufgerufen am 05.05.2023

Kapitel 3. Ergebnisse

3.2. Materialien

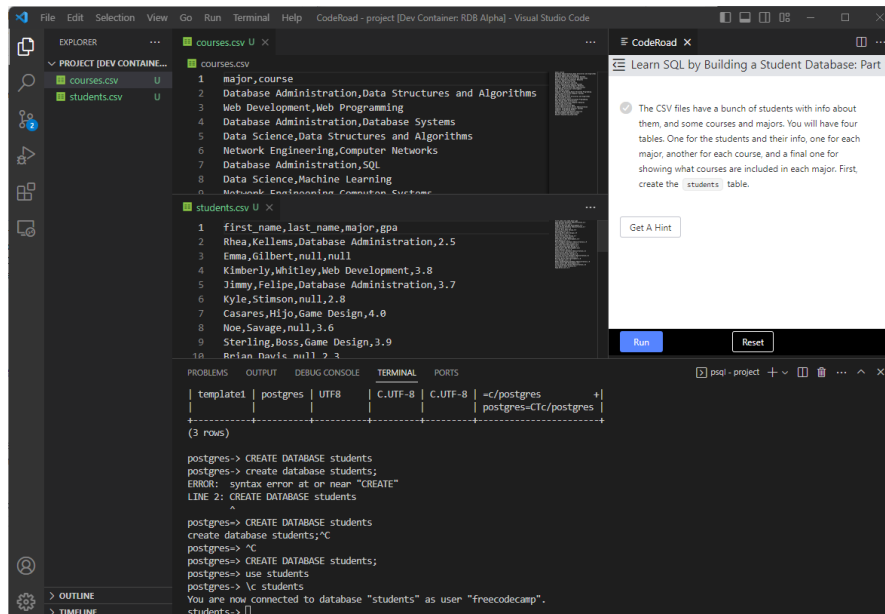


Abbildung 3.14: freeCodeCamp – „Learn SQL by Building a Student Database: Part 1“ eine SQL-Übung

Der gemeinnützige Verein *freeCodeCamp* bietet Interessierten eine große Anzahl an geführten, kostenlosen Lerninhalten rund um die Entwicklung von Anwendung an. Ziel ist es den Interessierten das nötige Fachwissen und die Praxis zu vermitteln, um einen Job als Softwareentwickler:in ausüben zu können. Auch bietet freeCodeCamp nach erfolgreichem Abschluss eines Lernplans ein Zertifikat an.

Ein Teil der Inhalte beschäftigt sich mit *relationalen Datenbanken*⁸. Die Abbildung 3.14 zeigt exemplarisch eine Aufgabe aus dem Kurs „Learn SQL by Building a Student Database: Part 1“, in der die essenziellen SQL Befehle gelehrt werden. Besonders interessant ist, dass die Studierenden mit einer echten PostgreSQL Instanz interagieren und daher direkt den Praxisbezug erwerben. Die Interaktion findet mit dem Terminal Frontend *psql* von PostgreSQL statt (siehe untere Hälfte der Abbildung 3.14). Im oberen rechten Drittel der Lernumgebung wird die Aufgabenstellung dargestellt, welche auch dafür genutzt wird, um Wissen zu vermitteln und anschließend abzufragen. Außerdem können die Studierenden sich mit dem Button „Get a Hint“ eine Hilfestellung für die Lösung der Aufgabe anzeigen lassen. Die Aufgaben sind aufeinander aufbauend, daher ist das Überspringen nicht möglich.

⁸freeCodeCamp’s Lehrplan für relationale Datenbanken <https://www.freecodecamp.org/learn/relational-database/>, aufgerufen am 10. Mai 2023

Im Gegensatz zu den rein webbasierten Materialien setzt **freeCodeCamp** für die Lehre von RDBMS zahlreiche Anforderungen an das System voraus. Um eine Lerneinheit zu starten, benötigen die Studierenden die Virtualisierungsumgebung *Docker* und den Code-Editor *VSCode* mit der *CodeRoad* Erweiterung. Durch die Virtualisierungslösung können die Studierenden Systemunabhängig dieselben Voraussetzungen vorfinden. Diese initiale Hürde haben auch die Entwickler von *freeCodeCamp* erkannt und bieten seit März 2022⁹ eine Version an, die den Code-Editor *VSCode* im Browser verfügbar macht und die erforderliche Virtualisierung auf einem zentralen System ausführt, d.h. ohne Installation auskommt.

3.2.6 Relationale Algebra

Essenzieller Bestandteil für die Abfragen einer relationalen Datenbank sind die Operationen, die auf die strukturierten Daten angewendet werden können. Mit diesen Operationen können die Datensätze unter anderem neu verknüpft, gefiltert und die Struktur verändert werden. Diese Operationen sind in der relationalen Algebra beschreibbar. Daher stellt die relationale Algebra die theoretische Grundlage für jede relationale DBMS dar. Die Lehre der relationalen Algebra findet normalerweise rein theoretisch statt. Allerdings kann durch den Einsatz interaktiver Materialien eine weitere Variation der Wissensvermittlung ermöglicht werden.

Mit dem Tool **RelaX** - **r**elational **a**lgebra **c**alculator der *University of Innsbruck* können Studierende interaktiv Abfragen in der relationalen Algebra entwickeln und ausführen. Nach der Ausführung der Abfrage wird das Ergebnis in tabellarischer Form dargestellt. Zugleich wird die Abfrage in einem Graphen visuell repräsentiert, um die einzelnen Operationen der Abfrage zu illustrieren (siehe Abbildung 3.15). In dem Graphen werden auch die Laufzeiten der einzelnen Operationen benannt. Hiermit können mögliche Auswirkung auf die Performance einer Abfrage nachvollzogen werden.

RelaX Ebenso können die Studierenden neben den Abfragen der relationalen Algebra eine SQL-Abfrage konzipieren, um das Ergebnis dieser analysieren zu lassen. In dem Ergebnis wird die SQL-Abfrage zudem in relationale Algebra übersetzt, welches dem Verständnis für die Syntax der relationalen Algebra behilflich sein kann.

Als Basis für die Abfragen sind in dem Tool mehrere Datenbanken mit den entsprechenden Schemas hinterlegt. Dabei stammen die Schemas aus der Fachliteratur und

⁹Forumbeitrag zur Veröffentlichung der webbasierten Version des Kurses relationale Datenbanken <https://forum.freecodecamp.org/t/running-the-relational-database-curriculum-in-your-browser/500231>, aufgerufen am 10. Mai 2023

Kapitel 3. Ergebnisse

3.2. Materialien

der Lehre an den Hochschulen. Auch können in dem Tool eigene Schemas erstellt und hochgeladen werden.

The screenshot shows the Relax tool interface. On the left is a schema tree with tables: Movies, Genres, and Persons. The Movies table has columns: Movie_ID (number), Title (string), ReleaseDate (date), Genre_ID (number), Budget (number), OpeningWeek (number), Profit (number), Runtime (number), Certificate (number), SequelOf (number), Distribution (string). The Genres table has columns: Genre_ID (number), Name (string). The Persons table has columns: Person_ID (number). The main area has tabs for 'Relational Algebra', 'SQL', and 'Group Editor'. The SQL tab is active, showing a query: `1 π Title, ReleaseDate, Name σ Budget > 20000000 (Movies ⋈ Movies.Genre_ID = Genres.Genre_ID Genres)`. Below the query is an 'execute query' button, 'Download' and 'History' links. A query execution diagram shows the flow: Movies (20 rows) and Genres (7 rows) are joined (⋈) to form a 20-row intermediate result. This is then filtered (σ) by Budget > 20000000 to produce another 20-row intermediate result. Finally, a projection (π) is applied to the columns Title, ReleaseDate, and Name, resulting in the final 20-row output. The execution time is 2 ms.

Abbildung 3.15: Relax – Beschreibung und Ergebnis eines Ausdrucks aus der Movie Datenbank

IRA Ein weiteres Tool, das für die Lehre von relationaler Algebra wurde von Dr. Henrik Mühe an der TUM entwickelt und nennt sich IRA (Interaktive Relationale Algebra). In diesem Material können die Studierenden, wie in dem vorher beschriebenen Tool Relax einen Ausdruck in der relationalen Algebra entwickeln und das Ergebnis anzeigen lassen. Das Ergebnis wird ausschließlich in tabellarischer Weise dargestellt. Im Gegensatz zu der Relax Anwendung nutzt das IRA eine festes Datenbankschema, das nicht verändert werden kann. Außerdem können keine SQL-Befehle weder ausgeführt noch übersetzt werden, wie dies in der Relax Anwendung der Fall ist. Diese Limitationen führen zu einer übersichtlichen Oberfläche, wie in Abbildung 3.16 zu sehen ist.

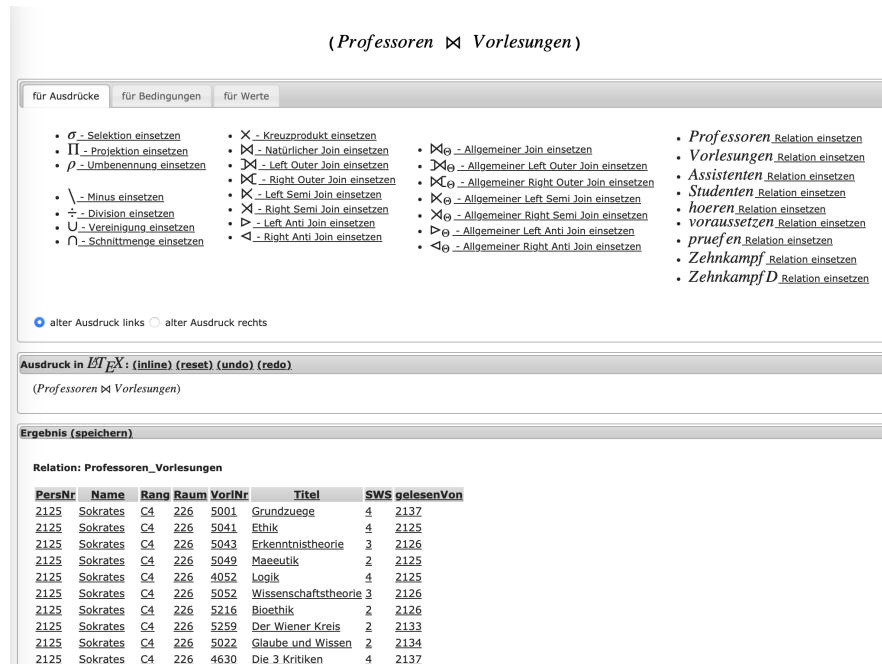


Abbildung 3.16: IRA – Eine Relation zwischen zwei Mengen in RA

3.2.7 Themenübergreifende Materialien

Themenübergreifende Materialien sind Materialien, die zu zwei oder mehr Themenbereichen Wissen vermitteln. Rund ein Viertel der recherchierten Materialien lassen sich als Themenübergreifende einstufen (siehe Abbildung 2.5). Im Folgenden wird ein repräsentatives Material, das in diese Rubrik eingeordnet wird, vorgestellt.

MCT-Trainer EILD.nrw Der MCT-Trainer wurde als Teil des EILD.nrw Projektes entwickelt. Dabei fundiert die Idee und die initiale Entwicklung auf einem Tool der Technischen Hochschule Köln¹⁰. Im Kontext des EILD.nrw Projektes wurde die Software zukuessive weiterentwickelt und als OER-Ressource veröffentlicht.

Mit dem MCT-Trainer sind die Studierenden in der Lage einen Test, basierend auf einem vorab ausgewählten Themengebiet, generieren zu lassen. Dieser nutzt dabei das Multiple-Choice-Verfahren, um die Inhalte zu prüfen. Bei einem Multiple-Choice Test wird eine Frage gestellt und der Prüfling erhält mehrere Antwortmöglichkeiten. Aus diesen Antwortmöglichkeiten müssen dann die Korrekten ausgewählt und markiert werden, dabei können mehrere Antworten zugleich richtig sein. Obwohl sich das Material auf Multiple-Choice Aufgaben fokussiert, werden auch Fragen gestellt, die im Freitext beantworten werden müssen. Insgesamt müssen 15 Fragen

¹⁰MCT im edb der TH Köln <https://edb2.gm.th-koeln.de/apps/mct/>

beantwortet werden, danach können die Studierenden die Musterlösung mit ihren eigenen Eingaben vergleichen.

Die Applikation kann auf ein großes Repertoire an Fragen zurückgreifen. Zum Zeitpunkt der Recherche waren in der Applikation 1317 Fragen hinterlegt, die auf 9 Kategorien aufgeteilt wurden¹¹.



Abbildung 3.17: MCT-Trainer – Eine Multiple Choice Frage aus dem Kapitel „Einführung in die Grundbegriffe der Datenbanken“

¹¹vgl. Fragenkatalog des MCT-Trainers <https://github.com/EILD-nrw/mct-trainer/blob/master/public/data/Questions.json>, aufgerufen am 10.05.2023

Kapitel 4

Fazit

Diese Recherche stellt interaktive Lehr- und Lernmaterialien vor, die in der Lehre von Datenbanksystemen eingesetzt werden können. Es wurden online verfügbare Materialien, als auch wissenschaftliche Arbeiten, die sich mit dem Entwurf oder der Evaluation solcher befassen, berücksichtigt. Letztendlich wurde nur ein Teil der insgesamt 59 gefundenen Materialien zur näheren Betrachtung ausgewählt.

Mit den gefundenen Materialien kann bereits eine Vielzahl der Themenbereiche aus DSE abgedeckt werden. Die Mehrheit der Materialien fokussiert sich nur auf einen Themenbereich mit spezifischen Aufgabestellungen (siehe Unterabschnitt 2.3.2). Vereinzelt wurden auch eigens entwickelte Umgebungen oder LMS gefunden, die gleich mehrere Themen umfassen. In der Aufteilung hinsichtlich der Themenbereiche ist zu erkennen, dass sich deutlich mehr Materialien auf SQL und ERM fokussieren, als auf z. B. Normalisierung, Relationale Algebra, etc. (vgl. Abbildung 2.4). Gleichzeitig ist festzustellen, dass die Materialien grundsätzlich für die Lehre von relationalen Datenbanken entworfen wurden. Andere Datenbanktypen wurden nicht thematisiert.

Die Materialien unterscheiden sich in ihrer Umsetzung. Der Großteil ist als Webanwendung implementiert. Allerdings gibt es auch hier Unterschiede in der Architektur. Während manche Materialien ihre Logik nur clientseitig ausführen, benötigen andere eine Verbindung zu einem externen Server. Ein Server wird häufig eingesetzt, wenn eine automatisierte Auswertung von Benutzereingaben vorgenommen wird und die Musterlösungen und Entscheidungslogik nur extern vorliegen. Historisch ältere Projekte setzen vermehrt auf Desktop-Anwendungen (z. B. KERMIT, SQL-Tutor) und wurden später teilweise auch ins Web portiert (z. B. EER-Tutor, SQL-Trainer). Der Großteil der Materialien kann als Übung eingesetzt werden. Ein nur geringer Teil davon vermittelt gleichzeitig auch das notwendige Wissen (vgl. Abbildung 2.2).

Die ersten Materialien wurden bereits Ende der 1990er entwickelt (siehe [HG98] und [Mit98]). Weil sie eigentlich für Studiengänge an den entsprechenden Universitäten entworfen wurden, stehen die Anwendungen inzwischen nicht mehr zur Verfügung. Hinzu kommen Anwendungen, die in LMS integriert wurden und daher einen Universitäts-Benutzeraccount voraussetzen. Diese sind nicht frei zugänglich.

Webbasierte Materialien sind zugänglicher für Studierende, weil sie darauf jederzeit systemunabhängig zugreifen können ohne zusätzliche Software installieren zu müssen. Auch die Bearbeitung der Aufgaben ist zeitlich nicht begrenzt, sodass Studierende ihr eigenes Lerntempo bestimmen können.

Auffällig ist, dass sich viele der vorgestellten Materialien in Teilen ähneln. Sowohl von der Zielsetzung als auch von der Umsetzung. Das liegt primär daran, dass es sich dabei um Weiterentwicklungen handelt (z. B. **EER-Tutor** von **KERMIT**) oder gleiche Technologien (z. B. **CBM**) oder Lernmethoden eingesetzt wurden.

Die Aufgaben oder Problemstellungen, die Studierenden in und mit den Anwendungen lösen sollen, müssen vorher von Lehrenden erstellt werden. Die Pflege von Aufgaben erfolgt entweder über ein Webinterface, auf das nur autorisierte Benutzer Zugriff haben, über die Verteilung von Dateien in speziellen Formaten, die dann in der Anwendung importiert werden können. Teilweise ist keine einfache Pflege von Inhalten möglich, da die Aufgaben und Musterlösungen im Quellcode hinterlegt sind.

Eine Ausnahme stellt **AutoER** dar, welches über eine Komponente zum dynamischen Erstellen von Aufgabenstellungen verfügt. Auch in [Geo08] wurde das automatische Erstellen von Aufgaben in Erwägung gezogen, auf Grund von unterschiedlicher Qualität der Ausgabe allerdings verworfen. Das fundamentale Problem bei automatisierten Aufgabenstellungen liegt darin, dass es nicht unbedingt eine einzige richtige Lösung gibt, sondern mehrere [siehe SM02, S. 377; Sol+10, S. 975]. Allerdings ist ein Vorteil von dynamisch generierten Inhalten, dass sie individuell an den Lernfortschritt von Studierenden angepasst werden können. Erfahrene Studierende können bspw. mit schwierigeren Aufgaben konfrontiert werden. Auch auf einzelne Problembereiche von Studierenden kann fokussierter eingegangen werden.

Ein weiterer Unterschied der Materialien ist die Herangehensweise und Zielsetzung hinsichtlich der Lehr- und Lernmethoden. In **NORMIT** müssen Studierende bspw. ihre Eingaben in Multiple-Choice Fragen rechtfertigen. Die Autoren argumentieren, dass dies eine effektive Lernmethode sei [vgl. Mit03b, S. 565]. Andere Lernmaterialien wiederum geben ausführliches Feedback zu den Eingaben und besonders einzelnen Komponenten. Die Qualität und der Detailgrad kann dabei variieren oder sich dynamisch anpassen (siehe Abschnitt 3.2.5, Abschnitt 3.2.1 und Abschnitt 3.2.3).

Grundsätzlich lassen sich die Materialien bezüglich des Anwendungsbereiches wie folgt aufteilen:

- Materialien können unterstützend zur Lehre eingesetzt werden, um bspw. Algorithmen und Datenstrukturen visuell zu veranschaulichen. Auch Abläufe, etwa die Zwischenschritte zur Normalisierung eines relationalen Schemas, können, wie in `Normalization Educational Tool` gezeigt, vereinfacht dargestellt werden.
- Einen ähnlichen Ansatz verfolgen auch solche Materialien, die neben dem Erklären von Konzepten das erlernte Wissen der Studierenden direkt prüfen, indem sie Aufgaben integrieren. Dazu zählt bspw. `SQLBolt` aber auch interaktive Vorlesungsfolien mit integrierten Quiz. `MonstER Park` gibt zwar keine direkte Einleitung in das Thema ERM, restriktiert die Benutzereingaben jedoch so, dass Studierende das Wissen spielerisch und ohne Vorerfahrung erlernen.
- Die dritte Art von Materialien dient lediglich als Übung für Studierende. Es wird Wissen über das Thema vorausgesetzt; Die Materialien selbst beinhalten diesbezüglich keinerlei Lehrinhalte. Durch Aufgaben können die Studierenden ihr Wissen vertiefen und ihre Herangehensweise üben.

Zur automatisierten Überprüfung wird in `KERMIT/EER-Tutor`, `NORMIT`, `SQL-Tutor`, `SQL-Trainer` und weiteren das Constraint-Based Modeling (CBM) eingesetzt [siehe SM02, S. 377; Mit03b, S. 570 f.; Mit03a, S. 176; Mit98, S. 310]. Bei diesem Verfahren wird die Eingabe anhand von vorher definierten Regeln (engl. „constraints“) ausgewertet. Eine richtige Lösung muss alle diese Regeln erfüllen. Dadurch muss sie nicht exakt der Musterlösung entsprechen. Gerade bei Problemen, die ergebnisoffen sind, also bei denen es keine einzelne korrekte Lösung gibt, kann trotzdem die Richtigkeit individuell analysiert werden [vgl. SM02, S. 386 f.; MS16, S. 451 f.]. Gleichzeitig ermöglichen genau diese Regeln das Generieren von komponentenspezifischem und detailliertem Feedback.

Bei einigen Materialien wird ebenfalls eine automatische Auswertung des Lernfortschritts vorgenommen. In `e-KERMIT` und `NORMIT` erhalten Studierende einen Überblick über ihren Lernfortschritt in verschiedenen Teilbereichen des ursprünglichen Problems. Fortschrittsbalken geben an, welches Unterthema bereits ausgiebig behandelt wurde oder wo die Studierenden noch Probleme haben. Dadurch werden Studierende motiviert weiter zu lernen [vgl. MS16, S. 452]. Lehrkräfte merken ebenfalls an, dass sie durch diese Übersicht gezielter auf einzelne Studierende eingehen konnten [siehe Sol+10, S. 978].

In nicht repräsentativen Experimenten wurde untersucht, ob sich der Einsatz interaktiver Lehr- und Lehrmaterialien positiv auf den Lernerfolg der Studierenden auswirkte. Die Auswertung wurde im Anschluss an die Zielsetzung und Umsetzung der jeweiligen Materialien durchgeführt. Das Vorgehen in den Experimenten ist ähnlich: Studierende wurden in zwei Gruppen aufgeteilt. Eine Kontrollgruppe musste entweder klassische Lernmethoden verwenden oder eine reduzierte Version der Materialien, wohingegen die andere Gruppe die vollumfänglichen Materialien nutzen durfte. In Tests, die vorher und nachher durchgeführt wurden, wurden die Ergebnisse beider Gruppen miteinander verglichen. Es sind Tendenzen zu erkennen, dass die Studierenden, die die neuen Materialien verwenden durften, in abschließenden Tests bessere Ergebnisse erzielten [vgl. Sol+10, S. 978; SM02, S. 386; Mit03b, S. 573 f.; FUL22, S. 630]. Allerdings sind die Ergebnisse häufig nicht aussagekräftig, weil das Experiment mit einer zu geringen Anzahl an Studierenden durchgeführt wurde (z. B. 62 Freiwillige in [SM02, S. 382], 48 in [Sol+10, S. 978]) und die Teilnahme oft freiwillig war. Zum letzteren Grund merken Mitrovic und Suraweera [MS16, S. 185 f.] an, dass dadurch die Ergebnisse verfälscht sein könnten.

Die Integration in bestehende bzw. bekannte LMS, wie etwa Moodle¹ oder Ilias², war keine Zielsetzung. Die Materialien wurden entweder direkt in bestehende proprietäre LMS integriert oder sind als eigenständige Anwendungen entworfen worden. Zwar bieten diese Umgebungen einige Features von LMS an (z. B. Benutzerverwaltung, Aufgabenverwaltung und -Auswertung, etc.), allerdings entsteht dadurch ein zusätzlicher Verwaltungsaufwand für Administratoren, diese Anwendungen zu betreiben. Unabhängige Materialien lassen sich über simple Mechanismen (z. B. `IFrames`) einbinden, bieten darüber hinaus jedoch keinerlei zusätzliche Informationen an das LMS.

¹Moodle Website: <https://moodle.org/>

²Ilias Website: <https://www.ilias.de/>

Kapitel 5

Ausblick

Die Rechercheergebnisse zeigen, dass es eine Vielzahl von Materialien für die DSE gibt. Daraus lässt sich ein genereller Bedarf ableiten. Durch Hochschulen werden die Forschung und Entwicklung neuer Materialien weiter vorangetrieben. Daher ist davon auszugehen, dass zukünftig weitere Materialien veröffentlicht werden. Basierend auf den bisherigen Forschungsergebnissen und Auswertungen können diese ständig verbessert werden und besser auf Studierende zugeschnitten werden. Dennoch gibt es Aufgabenstellungen, die in Zukunft gelöst werden müssen.

Die Recherche zeigt auch, dass sich viele Materialien auf bereits ausgiebig behandelte Themenbereiche, wie etwa SQL und ERM, fokussieren (siehe Unterabschnitt 2.3.3). Grundsätzlich gibt es gegen das Erproben neuer Lernmethoden und eine diverse Umsetzung nichts einzuwenden, es ist schließlich Teil der Forschung. Dennoch sind einige Themenbereiche schwächer bis gar nicht repräsentiert. Weil Evaluationen und Feedback von Studierenden gezeigt haben, dass mithilfe von interaktiven Lernmaterialien komplexe Inhalte besser verstanden werden, besteht hier drin ein großes Potenzial auch neue Themengebiete zu erschließen. Beispielsweise wurden Einblicke in die Implementierung von DBMS nicht thematisiert und werden nur als Teil klassischer Lehrinhalte angeboten.

Die Verwendung von Webtechnologien ermöglicht einen einfachen Zugang. Gerade zu Zeiten der weltweiten Corona-Pandemie, wo primär auf digitalen Unterricht gesetzt wurde, konnten Materialien online genutzt werden. Aus diesem Grund ist zu erwarten, dass auch zukünftige Materialien eine ähnliche Umsetzung wählen werden. Nicht zuletzt bestätigen diese Annahme alle kürzlichen Veröffentlichungen.

Auch LMS sind dadurch weiter in den Fokus gerückt. Sie werden an Schulen und Universitäten eingesetzt, um eine zentrale Plattform für den Wissensaustausch zu schaffen und Lehrinhalte anbieten zu können. Daran können interaktive Lehr-

und Lernmaterialien anknüpfen, indem sie in diese Plattformen integriert werden. Eine solche Anforderung konnte bisher nur in Einzelfällen festgestellt werden. Ein Vorteil wäre eine leichtere Integration in bestehende Lehrinhalte. Weiter können Lehrende eine Übersicht über den Lernfortschritt der Studierenden bekommen. Auf dieser Grundlage können Vorlesungen und Übungen entsprechend den aktuellen Defiziten dynamisch angepasst und für spätere Lehreinheiten aktualisiert werden. Allerdings setzt dies eine Kommunikation zwischen dem LMS und den interaktiven Lernmaterialien voraus.

Durch den Einsatz mehrerer Client-Server Anwendungen entsteht möglicherweise ein höherer Wartungsaufwand für Administratoren. Auf ältere Projekte konnte im Rahmen dieser Recherche bspw. nicht mehr zugegriffen werden, da die entsprechenden Server nicht mehr weiter betrieben werden.

Für die Verbreitung und Verwendung von freien Lehr- und Lernmaterialien müssen diese unter einer freien Lizenz stehen. In diesem Fall wird auch von OER gesprochen. Leider wurden viele der gefundenen Materialien nicht richtig oder überhaupt nicht gekennzeichnet. Dies stellt ein Problem für Lehrende dar, weil die Verwendung nicht rechtlich geregelt ist. Auch eine Anpassung oder Weiterentwicklung ist dadurch ausgeschlossen. Ganz im Sinne von OER sollten zukünftige Materialien immer mit entsprechender Software und Creative Commons Lizenz versehen werden.

Die Nachhaltigkeit solcher Materialien sollte eine wichtige Rolle spielen. Anstelle von immer neuen Implementationen, könnten Bestehende möglicherweise angepasst und weiterentwickelt werden. Durch Kollaboration mehrerer Stakeholder können so Erfahrungen und Wissen besser gebündelt werden. Voraussetzung ist, wie bereits angeschnitten, dass die Materialien und ggf. der Quellcode unter freier Lizenz veröffentlicht werden. Ein häufig vernachlässigter Aspekt ist die Sicherheit der Systeme und Anwendungen. Wichtige Sicherheitsupdates werden nicht ausgerollt oder als Aufgabe an die Administratoren übertragen. Auch ist auf die Anpassbarkeit zu achten. Jene Materialien, die nur für eine spezifische Lehrveranstaltung oder für konkrete Aufgabenstellungen entwickelt wurden, sind nicht für die allgemeine Verwendung geeignet.

Ein weiterer Ansatz, der in dieser Arbeit nicht betrachtet wurde, ist der Einsatz von CASE-Werkzeugen in der Lehre. Es kann auf bestehende Programme gesetzt werden, die bereits in der Softwareentwicklung erfolgreich angewendet werden. Soler et. al. [Sol+10, S. 974] argumentieren, dass diese häufig zu komplex für den Gebrauch in der Lehre sind. Mithilfe von entsprechendem Begleitmaterial können Studierende jedoch an diese Werkzeuge schrittweise herangeführt werden und erhalten gleichzeitig praktische Erfahrung.

Während der Recherche wurde deutlich, dass ein erheblicher Aufwand betrieben werden muss, um möglichst viele Materialien zu finden. Lehrende müssen also einen erheblichen Zeitaufwand betreiben, um geeignete Materialien entsprechend ihrer Anforderungen zu sichten. Aus diesem Grund wäre es sinnvoll, die Materialien an einer zentralen Stelle zu sammeln. Bereits existierende OER-Suchmaschinen verfolgen diesen Ansatz, fokussieren sich allerdings auf Lernmaterialien jeglicher Art und sind Themenübergreifend. Für das Fach Datenbanksysteme sind dort nur wenige Materialien indiziert.

Ähnliche Problemstellungen lassen sich auch auf andere Felder in der Informatik projizieren. Forschung und Ergebnisse aus diesen Bereichen könnten bei der Entwicklung weiterer Materialien für das Fach Datenbanksysteme berücksichtigt werden.

Literatur

- [ANJ19] Kanokwan Atcharyachanvanich, Srinual Nalintippayawong und Thanakrit Julavanich. „Reverse SQL Question Generation Algorithm in the DBLearn Adaptive E-Learning System“. In: *IEEE Access* 7 (2019), S. 54993–55004. DOI: 10.1109/ACCESS.2019.2912522.
- [Eri14] Erin Teo Yi Ling. *Teaching Algorithms with Web-based Technologies, Singapore: B. Comp.* <https://visualgo.net/fyp/erin-report.pdf>. 2014.
- [FUL22] Sarah Foss, Tatiana Urazova und Ramon Lawrence. „Automatic Generation and Marking of UML Database Design Diagrams“. In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*. SIGCSE 2022. Providence, RI, USA: Association for Computing Machinery, 2022, S. 626–632. ISBN: 9781450390705. DOI: 10.1145/3478431.3499376.
- [FWBS20] Heide Faeskorn-Woyke, Birgit Bertelsmeier und Jan Strohschein. „A Decision Tree Approach for the Classification of Mistakes of Students Learning SQL, a case study about SELECT statements“. In: *DELFI 2020 – Die 18. Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V.* Hrsg. von Raphael Zender, Dirk Ifenthaler, Thiemo Leonhardt und Clara Schumacher. Bonn: Gesellschaft für Informatik e.V., 2020, S. 211–216.
- [Geo08] Nikolay Georgiev. „A web-based environment for learning normalization of relational database schemata“. Magisterarb. Umeå, Schweden: Umeå University, Department of Computing Science, 2008.
- [Hal+12] Steven Halim, Zi Chun Koh, Victor Bo, H. H. Loh und Felix Halim. „Learning Algorithms with Unified and Interactive Web-Based Visualization“. In: Bd. 6. Olympiads in Informatics, 2012, S. 53–68.
- [HG98] Lynne Hall und Adrian Gordon. „A Virtual Learning Environment for Entity Relationship Modelling“. In: *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education*. SIGC-

- SE '98. Atlanta, Georgia, USA: Association for Computing Machinery, 1998, S. 345–349. ISBN: 0897919947. DOI: 10.1145/273133.274327.
- [Int16] International Organization for Standardization. *ISO 9075: Information technology — Database languages — SQL*. Standard. Geneva, CH, Dez. 2016.
- [Ish+22] Muhammad Ishaq, Adnan Abid, Muhammad Shoaib Farooq, Muhammad Faraz Manzoor, Uzma Farooq, Kamran Abid und Mamoun Abu Helou. „Advances in database systems education: Methods, tools, curricula, and way forward“. In: *Education and Information Technologies* 28.3 (2022), S. 2681–2725. DOI: 10.1007/s10639-022-11293-0.
- [KE15] Alfons Kemper und André Eickler. *Datenbanksysteme: Eine Einführung*. 10. Aufl. Oldenbourg Verlag, 2015. ISBN: 978-3-11-044375-2.
- [KS18] Anthony Kleerekoper und Andrew Schofield. „SQL Tester: An Online SQL Assessment Tool and Its Impact“. In: *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. ITiCSE 2018. Larnaca, Cyprus: Association for Computing Machinery, 2018, S. 87–92. ISBN: 9781450357074. DOI: 10.1145/3197091.3197124.
- [Kud15] Thomas Kudraß. *Taschenbuch Datenbanken*. 10. Aufl. Munich, Germany: Hanser, März 2015. ISBN: 978-3-446-44026-5.
- [Mit03a] Antonija Mitrovic. „An Intelligent SQL Tutor on the Web“. In: *Int. J. Artif. Intell. Ed.* 13.2–4 (2003), 173–197. ISSN: 1560-4292.
- [Mit03b] Antonija Mitrovic. „Supporting self-explanation in a data normalization tutor“. In: *Supplementary proceedings, AIED*. Bd. 2003. 2003, S. 565–577.
- [Mit98] Antonija Mitrovic. „Learning SQL with a Computerized Tutor“. In: *SIGCSE Bull.* 30.1 (1998), 307–311. ISSN: 0097-8418. DOI: 10.1145/274790.274318.
- [MS16] Antonija Mitrovic und Pramuditha Suraweera. „Teaching database design with constraint-based tutors“. In: *International Journal of Artificial Intelligence in Education* 26 (2016), S. 448–456.
- [Rak+09] Thomas C Rakow, Heide Faeskorn-Woyke, Bernhard Schiefer, Gottfried Vossen und Jürgen Wäsch. „Tools für die Lehre im Fach Datenbanken.“ In: *Datenbank-Spektrum* 9.29 (2009), S. 5–13.
- [Sch20] Johannes Schildgen. „MonstER Park-The Entity-Relationship-Diagram Learning Game.“ In: *ER Forum/Posters/Demos*. 2020, S. 150–157.
- [Shi20a] Shin-Shing Shin. „Structured Query Language Learning: Concept Map-Based Instruction Based on Cognitive Load Theory“. In: *IEEE Access* 8 (2020), S. 100095–100110. DOI: 10.1109/ACCESS.2020.2997934.

- [Shi20b] Shin-Shing Shin. „Structured Query Language Learning: Concept Map-Based Instruction Based on Cognitive Load Theory“. In: *IEEE Access* 8 (2020), S. 100095–100110. DOI: 10.1109/ACCESS.2020.2997934.
- [Shi22] Shin-Shing Shin. „Teaching Method for Entity-Relationship Models Based on Semantic Network Theory“. In: *IEEE Access* 10 (2022), S. 94908–94923. DOI: 10.1109/ACCESS.2022.3206028.
- [SK16] Christos Stefanidis und Georgia Koloniari. „An interactive tool for teaching and learning database normalization“. In: *Proceedings of the 20th Pan-Hellenic Conference on Informatics*. 2016, S. 1–4.
- [SM02] Pramuditha Suraweera und Antonija Mitrovic. „KERMIT: A constraint-based tutor for database modeling“. In: *Intelligent Tutoring Systems: 6th International Conference, ITS 2002 Biarritz, France and San Sebastian, Spain, June 2–7, 2002 Proceedings 6*. Springer. 2002, S. 377–387.
- [Sol+09] Josep Soler, Imma Boada, Ferran Prados, Jordi Poch und Ramon Fabregat. „Database design using a webbased e-learning tool“. In: *Proc. 2nd Workshop on Methods and Cases in Computing Education (MCCE)*. 2009, S. 31–40.
- [Sol+10] J. Soler, I. Boada, F. Prados, J. Poch und R. Fabregat. „A web-based e-learning tool for UML class diagrams“. In: *IEEE EDUCON 2010 Conference*. 2010, S. 973–979. DOI: 10.1109/EDUCON.2010.5492473.
- [TMS20] Faiza Tahir, Antonija Mitrovic und Valerie Sotardi. „Investigating the effects of gamifying SQL-Tutor“. In: *Virtual confernece: 28th International Conference on Computers in Education* (2020), S. 416–425.
- [UNE23] UNESCO. *Open Educational Resources*. Deutsch. 2023. URL: <https://www.unesco.de/bildung/open-educational-resources>.