

1. Praxistransferbericht

Konzeption und Implementierung einer Datenbank

Für das VR-Klassenzimmer

Name, Vorname: Zhang, Yüxi
Matrikelnummer: 684163
Ausbildungsbetrieb: Universität Potsdam
Studienjahrgang: 2021
Fachbereich: Duales Studium Wirtschaft • Technik
Studiengang: Informatik
Modul: 1-4021 PTB-TIT20A – 1. Praxistransferbericht
Betreuer Hochschule: Prof. Dr. Arthur Zimmermann
Betreuer Unternehmen: Axel Wiepke
Anzahl der Wörter: 2543

Vom Ausbildungsleiter zur Kenntnis
genommen:

16.03.2021 F.Züh
(Datum/Unterschrift)

26.03.2021 
(Datum/Unterschrift der/des Studierenden)

Abstract

Ziel des vorliegenden Praxistransferberichts ist die Konzeption und Implementierung einer Datenbank für das VR-Klassenzimmer Projekt der Universität Potsdam.

Für die Verständlichkeit der Arbeit werden die theoretischen Grundlagen von Datenbanken erläutert. Dazu gehören Aufbau, Funktionen, sowie Vor- und Nachteile einer Datenbank, das Datenbankmanagementsystem und das relationale Datenbankmodell, wobei andere Datenbankmodelle kurz erwähnt werden. Danach erfolgt eine kurze Erklärung des VR-Klassenzimmers, gefolgt von einer Anforderungsanalyse, wobei die Voraussetzungen und Ansprüche an die Datenbank beschrieben werden. Darauf aufbauend wird die Datenbank konzipiert und implementiert. Die Implementierung erfolgt mit dem Datenbankmanagementsystem „MySQL“ und der grafischen Benutzeroberfläche „phpMyAdmin“ auf einem Linux-Server der Universität Potsdam. Somit kann die Datenbank jederzeit von Axel Wiepke (Betreuer dieser Arbeit und wissenschaftlicher Mitarbeiter des Instituts für Informatik an der Universität Potsdam¹) und weiteren Kollegen genutzt werden. Zudem wurden beispielhaft drei SQL-Abfragen, auf Vorschlag von Herrn Wiepke, geschrieben und getestet. Es ist anzumerken, dass diese Datenbank noch nicht vollständig fertig ist, da nur Mock-Objekte (Beispieldaten bzw. Platzhalter für echte Objekte bzw. Daten) für diese Arbeit genutzt worden sind und später eventuelle Änderungen vorgenommen werden müssen. Dennoch kann die Datenbank in eine Programmiersprache integriert werden und damit in den Betrieb genommen werden.

¹<https://www.uni-potsdam.de/de/multimedia/team/mitarbeiter/axel-wiepke-m-sc>
Abrufdatum: 01.02.2021

Inhaltsverzeichnis

1	Einleitung	1
2	Theoretische Grundlagen von Datenbanken	2
2.1	Grundanforderungen von Datenbanksystemen	2
2.2	Relationales Datenbankmodell und Entity-Relationship-Modell.....	3
3	VR-Klassenzimmer.....	3
4	Anforderungsanalyse und Wahl des DBMS	5
5	Konzeption	5
6	Implementierung	6
7	Fazit und Ausblick.....	9
8	Literaturverzeichnis	10
9	Eidesstattliche Erklärung	11
10	Anhang.....	12

Abbildungsverzeichnis

Abbildung 1: Beispiel ERM in Chen-Notation, Rechtecke sind Entitäten, Rauten sind Beziehungen mit deren Beziehungsart, Ellipsen sind Attribute, Primärschlüssel sind unterstrichen, Fremdschlüssel nicht gekennzeichnet.....	3
Abbildung 2: Screenshot aus dem VR-Klassenzimmer mit Schülern in Gruppenarbeit.....	4
Abbildung 3: Ausgabe der 1. Abfrage.....	7
Abbildung 4: Ausgabe der 2. Abfrage, vor und nach der Umbenennung	7
Abbildung 5: Erklärung Berechnung der gelaufenen Distanz.....	7
Abbildung 6: Formel für Abstand zweier Punkte im zweidimensionalen Raum, d ist die Distanz	7
Abbildung 7: Ausgabe der 3. Abfrage.....	8
Abbildung 8: ERM nach Chen-Notation für die Datenbank des VR-Klassenzimmers	15
Abbildung 9: Beispiel: Erstellen der Tabelle "behaviour"	16
Abbildung 10: Beispiel Erstellen der Attribute für "behaviour", viele weitere Einstellungsmöglichkeiten wie z.B. Kommentare, Virtualität, Darstellungsumwandlung etc. möglich (Bild abgeschnitten).....	16
Abbildung 11: Beispiel SQL-Befehle von phpMyAdmin für Erstellung der Attribute für "behaviour", ohne phpMyAdmin müsste dieser Befehl manuell in die Kommandozeile eingetragen werden.....	16
Abbildung 12: Beispiel Erstellen einer Beziehung zwischen "behaviour" und "virtualStudent" mittels des PK bzw. FK "seatplace"	16
Abbildung 13: ERM in MySQL Workbench, Beziehungen anders als bei Chen-Notation, durch Auswahl eines PK werden alle FK farblich hervorgehoben.....	17
Abbildung 14: Abfrage 1	17
Abbildung 15: Abfrage 2, mit „COUNT“ Funktion.....	17
Abbildung 16: Für Abfrage 3: Prozedur "getDistance"	18
Abbildung 17: Für Abfrage 3: Funktion "calcDistance"	18

Quelle Abbildung 2: Axel Wiepke

Tabellenverzeichnis

Tabelle 1: fünf Beispiel Datensätze für 3. Abfrage.....	8
Tabelle 2: Beispiel ERM: Tabellen	12
Tabelle 3: Gegebene Mock-Daten (Attribute).....	13
Tabelle 4: angepasste Attribute für das ERM.....	14

Abkürzungsverzeichnis

DBMS - Datenbankmanagementsystem.....	2
DFG - Deutsche Forschungsgemeinschaft	1
ERM - Entity-Relationship-Modell	3
FK - Foreign Key (Fremdschlüssel)	3
GUI - graphical user interface	6
HWR - Hochschule für Wirtschaft und Recht.....	1
ID - Identifikationsnummer	3
kS - kein Schlüsselattribut	3
LoL - Lehramtsstudentin oder Lehramtsstudent	6
LuL - Lehramtsstudentinnen und Lehramtsstudenten	4
NFDI - Nationale Forschungsdateninfrastruktur.....	1
PK - Primary Key (Primärschlüssel)	3
SQL - Structured Query Language	3
SSH - Secure Shell	6
URL - Uniform Resource Locator	6
VR - Virtual Reality	4
vSoS - virtuelle Schülerin oder Schüler	13
vSuS - virtuelle Schülerinnen und Schüler.....	4

1 Einleitung

Wir leben in einer Informationsgesellschaft. Nach dem Aufstehen reicht ein Blick auf das Smartphone, um zu wissen, ob heute eine Regenjacke angemessen wäre oder nicht. Im Radio hören wir Staumeldungen während der Autofahrt, wodurch wir eine andere Route einschlagen können, um diesen zu umgehen. Die Zahl solcher „Informationsdienstleistungen“ hat in den letzten Jahren und Jahrzehnten ständig zugenommen und wird noch weiter zunehmen.“ Dabei spielt das Internet (World Wide Web) als Informationssystem eine entscheidende Rolle. [Zeh13] (S.9-19) Jeden Tag kommen sehr hohe Datenmengen in Umlauf und ohne Datenbanken wäre die Verarbeitung, Analyse, Auswertung und Darstellung von den Daten in unserer stets weiter digitalisierenden Welt nicht mehr möglich, da der Verwaltungsaufwand viel zu hoch wäre [Hal95] (S.1). Dies hat seine Vor- und Nachteile. Es ermöglicht uns zum Beispiel: etwas online zu bestellen, sich auf Social Media auszutauschen, in einer Cloud Daten zu speichern oder auf einer Streaming-Plattform einen Film zu schauen. Andererseits fallen aber auch bei bestimmten Seiten, wie z.B. Facebook, Nutzerdaten an [Bud17], welche in Datenbanken aufgenommen werden. Dadurch wird das Thema Datenschutz auch stetig relevanter. Für diejenigen, die solche Dienstleistungen nutzen, lohnt es sich also sich mit dem Thema der Datenspeicherung und Datenverwaltung auseinanderzusetzen.

Des Weiteren besteht eine deutliche Relevanz darin, dass Daten für Forschung und Lehre persistiert und weiterverwendet werden können. Diesbezüglich hat die Deutsche Forschungsgemeinschaft (DFG)² einen Kodex zum wissenschaftlichen Arbeiten verabschiedet. Damit wird die Qualität und Archivierung der Forschungsdaten sichergestellt und beim Aufbau einer Nationalen Forschungsdateninfrastruktur (NFDI)³ geholfen.

Diese Relevanz des Datenmanagement besteht auch in der Lehre, da diese helfen die Reflexion zu fördern und Kompetenzen und deren Zuwachs bewerten zu können.

Aus diesem Grund besteht der Bedarf für eine Datenbank, welche im Rahmen des Praxistransferberichtes der Hochschule für Wirtschaft und Recht (HWR) und des Instituts für Informatik der Universität Potsdam, für das VR-Klassenzimmer Projekt konzipiert und umgesetzt wird.

²<https://wissenschaftliche-integritaet.de/kodex/> Abrufdatum: 04.03.2021

³<https://www.bmbf.de/de/nationale-forschungsdateninfrastruktur-8299.html> Abrufdatum: 04.03.2021

2 Theoretische Grundlagen von Datenbanken

Datenbanken sind logisch zusammenhängende Sammlungen von Daten, die Ausschnitte der realen Welt beschreiben. Sie werden für spezielle Anwendungszwecke in einer Benutzergruppe genutzt. [Kud15] (S.20-21) Eine Datenbank wird meist in ein Datenbankmanagementsystem (DBMS) eingebettet. Zusammen ergeben sie ein Datenbanksystem.⁴ Meist wird es aber vereinfacht als Datenbank bezeichnet, da die Trennung meist nur für Datenbankentwickler eine Rolle spielen. [Kle16] (S.8) Das Datenbanksystem dient der Verwaltung und Organisation von Datenmengen auf einem elektronischen Medium.⁵ Ohne deren Einsatz kommt es oft zu einer unübersichtlichen Anzahl von Insellösungen, also Systemen, welche inkompatibel mit verwandten Systemen sind. Dabei werden Daten redundant, also mehrfach überflüssig in verschiedenen Bereichen, gespeichert. Dadurch entstehen ein hoher administrativer Aufwand und großer Speicherbedarf, was zu hohen Kosten führt, „die in keinem vernünftigen Verhältnis mehr zum Nutzen stehen.“ [Hal95] (S.1)

2.1 Grundanforderungen von Datenbanksystemen

Es gibt Grundanforderungen bzw. Aufgaben, die Datenbanksysteme erfüllen müssen. Einige der wichtigsten Anforderungen sind: die Persistenz, ein Datenbankschema, sowie das Einfügen, Ändern, Löschen und Lesen von Daten. Die Persistenz beschreibt, das Bestehen von Daten auf einem Speichermedium und dessen Wiederverwendbarkeit nach Beendigung der Software. Datenbankschemata wandeln Daten in Informationen um, wobei die Semantik eine entscheidende Rolle spielt. Als Beispiel kann die Zahl „32“ ausgegeben werden, jedoch kann erst mit der Zuweisung „Hausnummer“ daraus eine Information gewonnen werden. Die Daten sollten außerdem eingefügt, gelesen und gegebenenfalls geändert und gelöscht werden können. Wenn ein Softwaresystem diese Anforderungen erfüllt, kann sie Datenbank genannt werden. Allerdings sollten Datenbanksysteme noch weitere Funktionen aufweisen, in welchen das Datenbankmanagementsystem eine entscheidende Rolle spielt. Je nach DBMS und deren Anbieter existieren verschiedene und unterschiedlich viele Funktionalitäten. Einige Grundanforderungen jedes DBMS sind: redundanzfreie Datenhaltung (kein überflüssiges mehrfaches Speichern von Daten), parallele Nutzung der Daten, Rechteverwaltung, Datensicherung (Backup für technische Ausfälle und kritische Fehler) und Katalog (Überblick). [Kle16] (S.2-8) Das DBMS legt außerdem das Datenbankmodell fest, welches für die Datenbankentwicklung eine wichtige Rolle spielt. [Kud15] (S.21)

⁴Vgl. <https://www.oracle.com/de/database/what-is-database/> Abrufdatum: 02.02.2021

⁵Vgl. Ebd.

2.2 Relationales Datenbankmodell und Entity-Relationship-Modell

Ein Datenbankmodell ist die abstrahierte Darstellung der Daten und deren Beziehungen, wobei es Gesetzmäßigkeiten gibt, die sich von Modell zu Modell unterscheiden. [Gei14] (S.52)

Das relationale Datenbankmodell ist das meist verbreitete und kommerziell genutzte Modell. Es basiert auf Tabellen und Beziehungen und die Abfrage der Daten erfolgt über SQL (Structured Query Language) [Gei14] (S.60-63). Für die Konzeption einer relationalen Datenbank wird meist das Entity-Relationship-Modell (ERM) genutzt. In dieser Arbeit wird die Chen-Notation (Art der Darstellung des ERMs) genutzt. Eine Entität (Entity, Tabelle) ist ein eindeutig unterscheidbares Objekt, welches abstrakt (z.B. Rechnungen) oder real (z.B. Personen) sein kann. Diese Entitäten haben Eigenschaften (Datum, Alter, Namen), auch Attribute genannt. Zudem sollte jede Entität mindestens eine Beziehung (Relation) zu einer anderen Entität haben, welche mittels Primär- und Fremdschlüssel realisiert werden. Der Primärschlüssel (Primary Key, PK) ist ein eindeutiges Attribut für jeden Datensatz (Zeile) der Tabelle. Dafür wird meist etwas, wie eine Identifikationsnummer (ID) genutzt. Der Fremdschlüssel (Foreign Key, FK) (PK der 1. Tabelle) wird in eine andere Tabelle eingefügt und referenziert somit auf die 1. Tabelle. Des Weiteren existieren drei Arten von Beziehungen: 1:1, 1:N, N:M. Bei der 1:1 Beziehung steht genau ein Datensatz der 1. Tabelle zu einem Datensatz der 2. Tabelle in Relation. Entsprechend steht bei der 1:N Beziehung ein Datensatz der 1. Tabelle zu beliebig vielen Datensätzen der 2. Tabelle in Relation und bei N:M beliebig viele zu beliebig vielen (als Beispiel siehe Abbildung 1 S.3 und Anhang: Tabelle 2 S.12). Jedoch reicht ein ERM nicht aus, um alle Redundanzen zu beheben. Folglich werden Normalisierungen von Relationen (Normalformen) benötigt. Es gibt fünf Normalformen (über die sechste wird diskutiert), wobei jedoch nur die ersten drei für diese Arbeit eine Relevanz haben. Die darauffolgenden Normalformen setzen die vorherige Normalform als Bedingung voraus. Die 1. Normalform wird erreicht, wenn alle Attribute (kein Schlüsselattribut - kS) nur atomare (einfache) Werte enthalten und ein eindeutiger Primärschlüssel gegeben ist. Bei der 2. Normalform muss jedes Attribut (kS) vom PK voll funktional abhängig sein. Schließlich darf bei der 3. Normalform kein Attribut (kS) vom PK transitiv abhängen, d.h. kein Attribut (kS) darf abhängig von einem anderen Attribut sein (kS). [Sch17] (S.58-82)

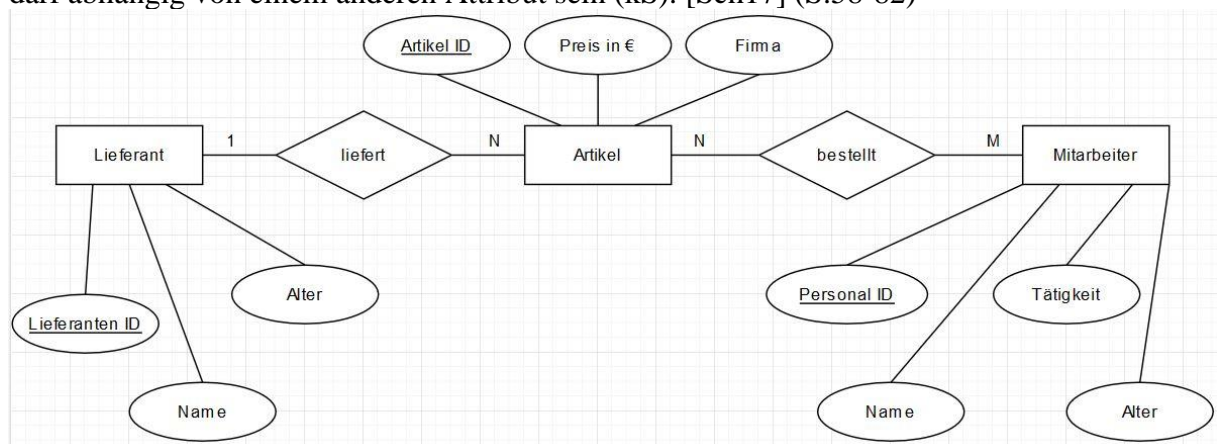


Abbildung 1: Beispiel ERM in Chen-Notation, Rechtecke sind Entitäten, Rauten sind Beziehungen mit deren Beziehungsart, Ellipsen sind Attribute, Primärschlüssel sind unterstrichen, Fremdschlüssel nicht gekennzeichnet.

3 VR-Klassenzimmer

3 VR-Klassenzimmer

Virtual Reality (Virtuelle Realität oder VR) ist eine dreidimensionale computergenerierte Umgebung. Dabei wird das Ziel verfolgt, eine Situation nachzubilden, mit welcher der Nutzer (User) interagiert. Mittels gezielten Feedbacks und Reflexion lässt sich dadurch Wissen aneignen. [Lac20] (S.76) Diese Umgebung kann über ein Head-Mounted-Display (sogenannte VR-Brille) dargestellt werden.

Das VR-Klassenzimmer ist ein Projekt von Axel Wiepke, Raphael Zender, sowie Eric und Dirk Richter [Wie19]. Das Projekt dient dem Verhaltenstraining von Lehramtsstudentinnen und Lehramtsstudenten (LuL) und soll ihnen die Möglichkeit bieten sich auf den späteren realen Unterricht vorzubereiten. Dabei üben sie, praxisnah und zielgerichtet auf Unterrichtsstörungen einzugehen. Es wird ein Klassenzimmer nachgebildet, worin sich virtuelle Schülerinnen und Schüler (vSuS) befinden (siehe Abbildung 3 S.4).



Abbildung 2: Screenshot aus dem VR-Klassenzimmer mit Schülern in Gruppenarbeit.

Die vSuS sind optisch in der zweiten Sekundarstufe und ethnisch divers. Im Klassenzimmer sind verschiedene Anordnungen der Tische möglich. Diese vSuS werden von einem Coach / einer Coachin über eine Weboberfläche gesteuert, wobei gezielte Aktionen von den vSuS ausgelöst werden. Die Aktionen der vSuS basieren auf realen Unterricht in einer Klasse und werden nach dem Buch „Techniken der Klassenführung“ [Kou06] kategorisiert. Zum einen wird die Mitarbeit, zum anderen auch das Störverhalten (z.B. den Nachbarn schlagen oder einen Papierball werfen) im Unterricht simuliert. Die LuL sollen entsprechend auf dieses Verhalten eingehen und dadurch ihre Klassenmanagementkompetenzen unter Beweis stellen und gegebenenfalls verbessern. Der Student kann real im 3x3 Meter großen (Play Area) Raum laufen, wodurch sich der Charakter im VR-Klassenzimmer auch bewegt. Jedoch ist dieser Raum begrenzt, deshalb werden Teleportation mittels des Controllers genutzt. Durch die Daten, die bei einer Sitzung des VR-Klassenzimmers entstehen, lassen sich die Handlungen des Studenten genau beschreiben und evaluieren, wobei der Fokus auf den Unterrichtsstörungen liegt. Aufnahmen unterstützen diesen Prozess und ermöglichen es kritische Szenen erneut anzuschauen. Somit können erlernte Kompetenzen gezielt angewendet und reflektiert werden. [Wie19]

4 Anforderungsanalyse und Wahl des DBMS

Eine Datenbank wird meist nur für einen bestimmten Anwendungsfall konzipiert, wie in diesem Fall für das VR-Klassenzimmer. Dabei gibt es einige allgemeine Anforderungen, die schon früher in dieser Arbeit genannt worden sind (siehe 2.2 Datenbankmanagementsystem).

Diese Datenbank soll jedoch weiterhin auch auf einem Linux-Server der Universität Potsdam laufen. Dabei spielen die Effizienz, Flexibilität, Kosten, das Betriebssystem, Sicherheit (Updates), Benutzerfreundlichkeit, die aktuelle Infrastruktur des Betriebs und viele weitere Faktoren eine Rolle bei der Auswahl des DBMS (welches auch das Datenbankmodell festlegt (siehe 2.1 Grundanforderungen von Datenbanksystemen)).

Die Wahl fiel auf das DBMS: MySQL, da es viele wichtige Anforderungen erfüllt.

Es ist ein relationales DBMS, welches für diese Datenbank bevorzugt wird, weil eine Tabellendarstellung für die von Herrn Wiepke gegebenen Mock-Daten (siehe 5. Konzeption) geeignet ist. Darüber hinaus nutzt das Institut für Informatik der Universität Potsdam in ihrer Infrastruktur MySQL-Datenbanken aufgrund der Benutzerfreundlichkeit, kostenlosen Nutzung und Kompatibilität mit Linux-Servern. Die leichte Handhabung ist darauf begründet, dass die Abfragesprache SQL für viele Mitarbeiter schon bekannt ist und es ein sehr verbreitetes DBMS ist, wodurch viele Sicherheits- und Funktionsupdates durchgeführt werden. [Her18] (S. 5-8)

5 Konzeption

Die von Herrn Wiepke übergebenen Mock-Daten weisen Informationslücken auf, wobei deren genaue Funktion und Bedeutung nicht klar definiert wurden. Über mehrere Gespräche wurde eine Tabelle angefertigt, welche diese erklären (siehe Anhang: Tabelle 3 S.13). Allerdings wiesen diese Daten viele Redundanzen auf, wie zum Beispiel: MisbehaviourCount und LevelOfMisbehaviour, da sich aus der Anzahl der Änderungen von LevelOfMisbehaviour schon die generelle Anzahl der Fehlverhalten ausrechnen lässt. Zudem wurde die Normalisierung noch nicht durchgeführt, wie z.B. bei evaluationMap. Hier treten mehrere Werte auf, weshalb sie nicht atomar sind. Des Weiteren fehlen alle Primärschlüssel, Beziehungen und weitere wichtige Attribute, um das ERM vollständig erstellen zu können. Nach Absprache mit Herrn Wiepke wurden Redundanzen verringert und die Normalisierung angewendet (siehe Anhang: Tabelle 4 S.14). So konnte das ERM erstellt werden (siehe Anhang: Abbildung 8 S.15).

6 Implementierung

Die Implementierung erfolgt auf einem Linux-Server des Instituts für Informatik der Universität Potsdam. Die Vorgehensweise erfolgt nach Finn-Lucas Ziehe (meinem Ausbilder und Systemadministrator am Institut für Informatik⁶). Zuerst wurde auf einem Test Server, die Datenbank getestet, um den Betrieb des produktiven Servers nicht zu beeinträchtigen. Nach dem Einloggen auf dem Server per SSH (Secure Shell), wodurch Zugriff auf das Betriebssystem gewährt wird, müssen bestimmte Anwendungen für die Datenbank installiert werden. Bei Linux können Installationen über das Terminal (Kommandozeile oder Konsole) geregelt werden, da Installationspakete vorliegen. Die Befehle für die Installation werden am Ende von der Implementierung gelistet. Apache 2.4 ist ein Webserver, wodurch auf die Webseite, mittels IP (z.B. „141.89.54.121“) oder URL (z.B. „beispiel.de“) zugegriffen werden kann. Danach wird „MySQL“ und „phpMyAdmin“ installiert. „phpMyAdmin“⁷ ist eine webbasierte grafische Benutzeroberfläche (GUI, graphical user interface), wodurch die Handhabung der Datenbank nicht über das Terminal geregelt werden muss. So wurde die zuvor konzipierte Datenbank erstellt (siehe Anhang: Abbildung 9-12 S.16). Zusätzlich wurde MySQL Workbench⁸ genutzt, um das ERM darzustellen (siehe Anhang: Abbildung 13 S.17). Die Datenbank wurde getestet und exportiert, wobei alle Befehle, die unter phpMyAdmin ausgeführt wurden, in ein Skript geschrieben werden. Dieses wurde dann auf dem richtigen Server namens: „Echalk“ ausgeführt und die Datenbank kann in den Betrieb genommen werden.

Linux-Befehle für Installation (ohne Anführungsstriche)

Apache: „sudo apt install apache2“

MySQL: „sudo apt install mysql-server“, Installation des DBMS in Server Ausführung

phpMyAdmin: „sudo apt install phpmyadmin“, dann apache2 auswählen und ein Passwort erstellen

Beispiele für SQL-Abfragen

Herr Wiepke hat einige Beispiele von SQL-Abfragen genannt, die später bei der richtigen Datenbank relevant sein können. Die Abfragen können über eine Auswahl auch auf phpMyAdmin gespeichert werden und später abgerufen werden. Hierbei ist zu beachten, dass nur ein Datensatz implementiert worden ist, daher können die Ausgaben bei mehreren Datensätzen anders aussehen (außer bei Abfrage 3).

Anhand der Matrikelnummer sollen folgende Werte während einer Sitzung des VR-Klassenzimmers abgerufen werden können:

1. die Anzahl der Schüler
2. die Anzahl der Fehlverhalten
3. die gelaufene Distanz der Lehramtsstudentin oder des Lehramtsstudenten (LoL)

Zu 1.

⁶ <https://www.uni-potsdam.de/de/multimedia/team/sekretariatundtechnik/finn-lucas-ziehe>
Abrufdatum: 07.03.2021

⁷ <https://www.phpmyadmin.net/> Abrufdatum: 08.03.2021

⁸ <https://www.mysql.com/de/products/workbench/> Abrufdatum: 08.03.2021

6 Implementierung

Siehe Anhang: Abbildung 15 (S.18)

Die Abfrage gibt die Matrikelnummer und jede Anzahl der Studenten und Sitzungs-ID aus, wo die/der LoL teilgenommen hat (siehe Abbildung 3 S.7).

matriculationNumber	studentCount	session-ID
689576	30	3

Abbildung 3: Ausgabe der 1. Abfrage

Zu 2.

Siehe Anhang: Abbildung 16 (S.18)

Die Abfrage zählt die Anzahl der Einträge von levelOfMisbehaviour einer Sitzung, wo die/der LoL teilgenommen hat. Für die Anschaulichkeit wurde es umbenannt (siehe Abbildung 4 S.7).



Abbildung 4: Ausgabe der 2. Abfrage, vor und nach der Umbenennung

Zu 3.

Für die Erklärung der Berechnung (siehe Abbildung 5 S.7). Die roten Sterne bilden jeweils die Position (einen Punkt) der/des LoL zu einem bestimmten dataCount (Zeitpunkt) ab und die roten Pfeile jeweils dem Abstand von einem dataCount zum anderen. Um die gesamte Strecke zu berechnen die die/der LoL zurückgelegt hat, muss der Abstand zwischen den dataCounts summiert werden.

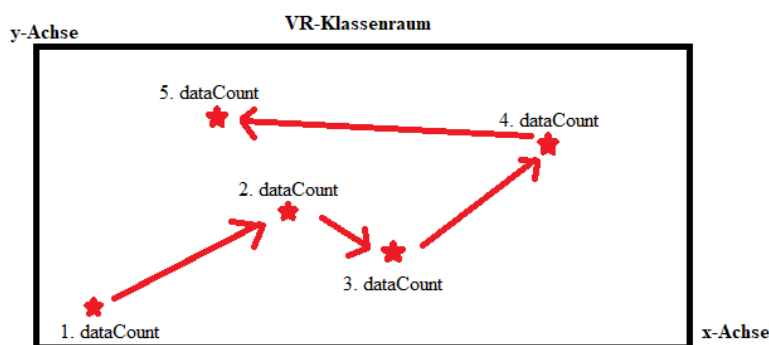


Abbildung 5: Erklärung Berechnung der gelaufenen Distanz

Der Abstand zweier Punkte errechnet sich aus dieser Formel:

$$d := \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Abbildung 6: Formel für Abstand zweier Punkte im zweidimensionalen Raum, d ist die Distanz

6 Implementierung

Zudem werden für die Anschaulichkeit weitere Mock-Daten für die Entität „movement“ in die Datenbank eingefügt, da sich aus einem Datensatz (einer Position) keine Abstände berechnen lassen. Für das Beispiel werden fünf Datensätze implementiert (selbst erfunden, nicht nach Abbildung) (siehe Tabelle 1 S.8)

dataCount	positionX	positionY	timestamp	trainee-ID
0	0	0	1615303964	1
1	4.03	-5.47	1615303984	1
2	5.63	-4.78	1615304004	1
3	7.05	-6.92	1615304024	1
4	6.55	-6.55	1615304044	1

Tabelle 1: fünf Beispiel Datensätze für 3. Abfrage

Daraus ist folgende Abfrage entstanden: siehe Anhang: Abbildung 17-18 (S. 19)

Für die Umsetzung der Abfrage wurde eine Prozedur „getDistance“ und eine Funktion „calcDistance“ geschrieben. „getDistance“ zählt die Anzahl der Datensätze und wandelt diese in Anzahl von Iterationen (wiederholte Anweisung in einer Schleife) um. Nach jeder Iteration werden zwei Zeilen (nur „positionX“ und „positionY“) der Tabelle „movement“ gewählt und umbenannt. Bei der ersten Iteration werden „positionX“ und „positionY“ bei dataCount = „0“ in „x1“, „y1“ umbenannt und bei dataCount = „1“ in „x2“, „y2“. Die umbenannten Daten werden als Parameter in die Funktion „calcDistance“ eingefügt, wo die oben genannte Formel für die Abstandberechnung zwischen beiden Positionen ausgeführt wird und werden dann zwischengespeichert. Bei der zweiten Iteration wird um eins hochgezählt, also wird die Zeile mit dataCount = „1“ ausgewählt und in „x1“, „y1“ und die Zeile mit dataCount „2“ wird in „x2“, „y2“ umbenannt und wieder nach der Berechnung zwischengespeichert. Dies wird abhängig von der Anzahl der Iterationen wiederholt. So wird der Abstand jeder Position zu der nächsten Position berechnet und abgespeichert. Zuletzt werden diese Abstände in „getDistance“ addiert und ausgegeben. Dieser Wert spiegelt die gesamte gelaufene Distanz wieder (siehe Abbildung: 8 S.8).

walkedDistance

11.727

Abbildung 7: Ausgabe der 3. Abfrage

7 Fazit und Ausblick

Dieser Praxistransferbericht hat das Thema Datenbanken und das VR-Klassenzimmer nur grob angeschnitten. Trotzdem ist zu erkennen, dass Datenbanken eine erhebliche Relevanz in der heutigen Datenwelt spielen und das Thema sehr komplex erscheinen kann. Jedoch ist es möglich sich durch die Einarbeitung gut in die Thematik hineinzufinden, wodurch die Konzeption und Implementierung der Datenbank auf dem Server gelungen sind. Dies ist auch auf die Wahl des DBMS „MySQL“ zurückzuführen, welches sehr bekannt ist und dadurch eine gute Dokumentation von Informationen vorliegt. Außerdem wird „MySQL“ wird auch in der Infrastruktur des Institut für Informatik verwendet, wodurch die Datenbank ohne Probleme weiter von Mitarbeitern genutzt und verwaltet werden kann. Die Konzeption gelang aber nicht ohne Schwierigkeiten, wie z.B. das Erstellen des ERMs, da die gegebenen Mock-Daten viele Informationslücken aufwiesen und viele Redundanzen sich erst mit größeren Zeitaufwand verringern ließen. Zudem wurden für diese Arbeit Mock-Daten nur für Anschauungszwecke verwendet. Folglich müssen eventuelle Änderungen an der Datenbank durchgeführt werden, was bei zukünftigen Erweiterungen der Datenbank unausweichlich sein wird. Eine Zertifizierung (SSL-Zertifikat) der Webseite der Datenbank wäre ebenfalls wichtig, da alle Nutzer der Datenbank dadurch eine verschlüsselte bzw. sichere Verbindung mit dem Server aufbauen können. Zusammenfassend, kann die konzipierte und implementierte Datenbank kann jedoch schon in eine Programmiersprache integriert und somit in den Betrieb genommen werden.

8 Literaturverzeichnis

- [Bud17] Oliver Budzinski und Marina Grusevaja: *Die Medienökonomik personalisierter Daten und der Facebook-Fall*, Working Paper. 2017. Abgerufen 28. Feb. 2021. [online] <http://hdl.handle.net/10419/168318>
- [Gei14] Frank Geisler: *Datenbanken: Grundlagen und Design*. mitp Verlag, 2014.
- [Hal95] Anton Hald und Wolf Nevermann: *Datenbank-Engineering für Wirtschaftsinformatiker: Eine praxisorientierte Einführung*. Vieweg, 1995.
- [Her18] Frank Herrmann: *Datenorganisation und Datenbanken. Praxisorientierte Übungen mit MS Access 2016*. Wiesbaden: Springer Vieweg, 2018.
- [Kle16] Stephan Kleuker: *Grundkurs Datenbankentwicklung*. Wiesbaden: Springer Vieweg, 2016.
- [Kou06] Jacob S. Kounin: *Techniken der Klassenführung. Standardwerke aus Psychologie und Pädagogik. Reprints..* Münster: Waxmann, 2006.
- [Kud15] Thomas Kudraß: *Taschenbuch Datenbanken*. Carl Hanser, 2015.
- [Lac20] Maximilian Lackner und Horst Orsolits: *Virtual Reality und Augmented Reality in der Digitalen Produktion*. Wiesbaden: Springer Gabler, 2020.
- [Sch17] Edwin Schicker: *Datenbanken und SQL: Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL*. Wiesbaden: Springer Vieweg, 2017.
- [Wie19] Axel Wiepke, Eric und Dirk Richter und Raphael Zender: *Einsatz von Virtual Reality zum Aufbau von Klassenmanagement-Kompetenzen im Lehramtsstudium*. N. Pinkwart und J. Konert (Hrsg.). DELFI 2019. Bonn: Gesellschaft für Informatik e.V.: 2019, S. 133-144. Abgerufen 12. Jan. 2021. [online] <https://dl.gi.de/handle/20.500.12116/24390>
- [Zeh13] Carl A. Zehnder: *Informationssysteme und Datenbanken*. Stuttgart: B. G. Teubner, 2013.

9 Eidesstattliche Erklärung

Ich erkläre ehrenwörtlich:

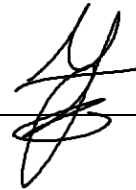
1. dass ich meinen Praxistransferbericht selbständig verfasst habe,
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe,
3. dass ich meinen Praxistransferbericht bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

26.03.2021

Ort, Datum

Yüxi Zhang



10 Anhang

10 Anhang

Lieferant		
<u>Lieferanten ID</u>	Name	Alter
Liefer2123	Keanu Reeves	23
Liefer1234	Justin Bieber	32
Liefer2314	Kevin Wolter	23
Liefer5614	Ben Tatum	19

Artikel			
<u>Lieferanten ID</u>	<u>Artikel ID</u>	Preis in €	Firma
Liefer2123	P1	10.90	World of Pizza
Liefer2123	B1	8.90	Jim Burger
Liefer2123	P3	10.90	World of Pizza

Mitarbeiter			
<u>Personal ID</u>	Name	Tätigkeit	Alter
YZ123	Yuexi Zhang	Auszubildener	19
BL641	Ben Lucken	Auszubildener	20
MZ789	Maria Zieher	Professorin	42

bestellt	
<u>Personal ID</u>	<u>Artikel ID</u>
YZ123	P1
BL641	B1
MZ789	P3

Tabelle 2: Beispiel ERM: Tabellen

Durch das Einfügen des Fremdschlüssels von Lieferant in Artikel und den Vereinfachungsregeln von 1:N Beziehungen, muss die Relation „liefert“ nicht als eigene Tabelle aufgeführt werden, da der Fremdschlüssel „Lieferanten ID“ in der Tabelle (Entität) Artikel aufgeführt wird und somit auf den Lieferanten verweist.

10 Anhang

Attribut	Beispiel (ein Datensatz)	Bedeutung
StudentCount	30	Anzahl Schüler im VR-Klassenzimmer
ChosenScene	2	1 ist Frontalunterricht, 2 ist Gruppenarbeit (Tischaufbau)
LevelOfMisbehavior	0	0 (kein), 1 (geringes), 2 mittleres, 3 (starkes) Fehlverhalten
NumberOfExperiment	0	Verhaltensskript für virtuelle Schülerin oder Schüler (vSoS), 0-5 möglich
isPresentation	True	Start der Stunde: True (Schüler schauen LoL an), False (Schüler schreiben)
isAutomaticIntervention	True	Skript Beendigung Fehlverhalten: True (Automatisch durch Nähe der/des LoL), False (durch manuelles Klicken auf Schüler)
isNonScripted	False	Skript: True (zufälliges Fehlverhalten), False (gesteuertes Fehlverhalten)
isExperiment	False	Skript (Erzwis) für Verhalten der Schüler: True (Nutzen Skript), False (andere Steuerung)
evaluationMap	1,4.03,-5.47	Position von LoL mit DataCount (Zählintervall für Position), PositionX, PositionY
walkedDistance	0	Gelaufene Distanz der/des LoL
MisbehaviourCount	0	Anzahl der gezählten Fehlverhalten
MisbehaviourSolved	0	Anzahl Übergang von Störung in positives Verhalten (durch Coach oder LoL)
MisbehaviourSeen	0	Störung in Sichtbereich der/des LoL oder durch manuelle Markierung
repetitionCount	0	Anzahl der allgemeinen VR-Klassenzimmer Sitzungen

Tabelle 3: Gegebene Mock-Daten (Attribute)

10 Anhang

Attribut	Beispiel (ein Datensatz)	Bedeutung
levelOfMisbehaviour	0	0 (kein), 1 (geringes), 2 mittleres, 3 (starkes) Fehlverhalten
nameOfBehaviour	watchTeacher	PK für Behaviour, Verhalten des vSoS, hier: LoL anschauen
name	Anna	Name des vSoS
seatplace	01L	PK für Virtual Student, Sitzplatz für Schüler
chosenScene	2	1 ist Frontalunterricht, 2 ist Gruppenarbeit (Tischaufbau)
studentCount	30	Anzahl Schüler im VR-Klassenzimmer
session-ID	3	PK für VR-Classroom, einzigartige ID für Sitzung
matriculationNumber	689576	Matrikelnummer einer/eines LoL
trainee-ID	1	PK für Trainee Teacher, einzigartige ID für LoL
dataCount	1	Zeitpunkt Aufnahme Daten (z.B. alle 10 Frames)
positionX	4.03	X-Wert der Position der/des LoL zum Data Count
positionY	-5.47	Y-Wert der Position der/des LoL zum Data Count
timestamp	1615208795	Zeitangabe in Linux (Mon Mar 08 2021 14:06:35 GMT+0100)
interactionName	lookedAt	Interaktion von LoL mit Schüler, hier: guckt Schüler an
interaction-ID	11	PK für Interaction, einzigartige ID für Interaktion
state-ID	11	PK für Class-State, einzigartige ID für Status

Tabelle 4: angepasste Attribute für das ERM

10 Anhang

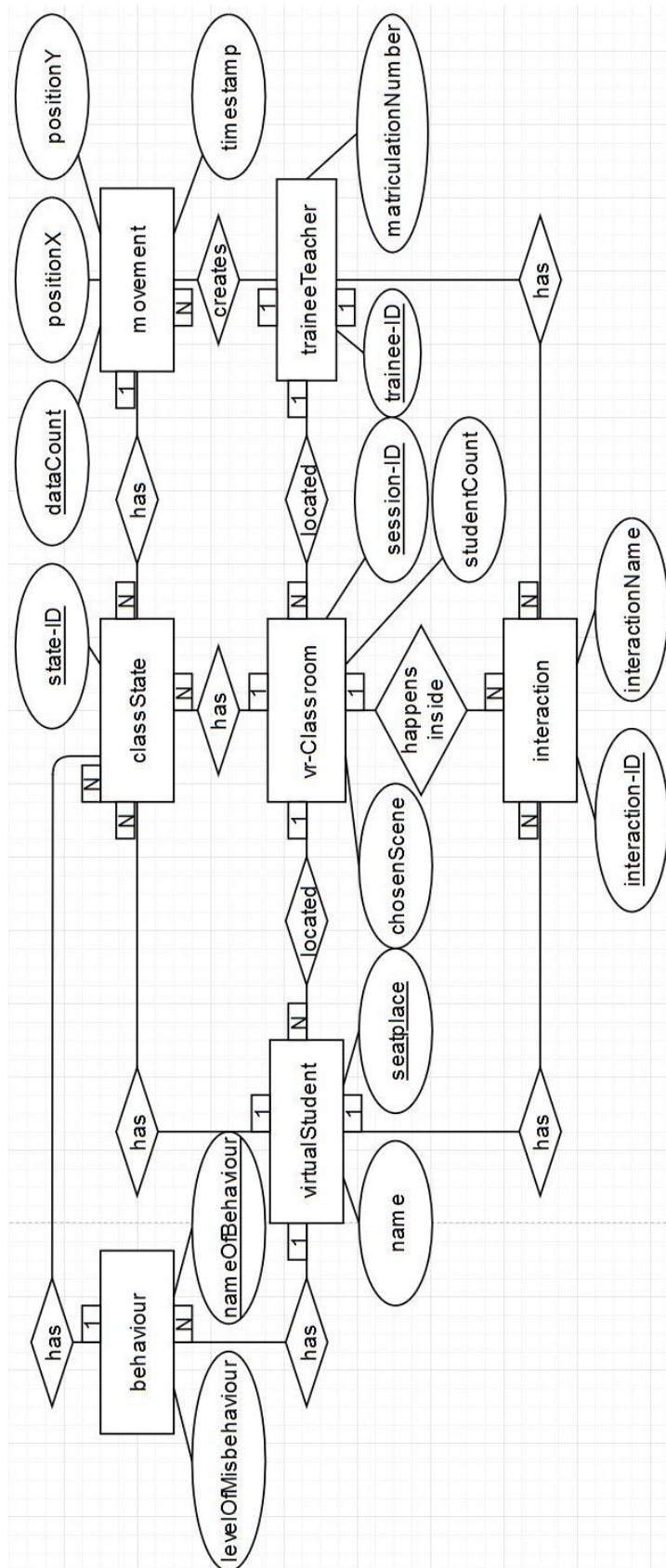


Abbildung 8: ERM nach Chen-Notation für die Datenbank des VR-Klassenzimmers

10 Anhang

Abbildung 9: Beispiel: Erstellen der Tabelle "behaviour"

Abbildung 10: Beispiel Erstellen der Attribute für "behaviour", viele weitere Einstellungsmöglichkeiten wie z.B. Kommentare, Virtualität, Darstellungsumwandlung etc. möglich (Bild abgeschnitten)

```
CREATE TABLE `TEST`.`behavior` ( `levelOfMisbehaviour` INT(1) NOT
NULL COMMENT '0 (kein), 1 (geringes), 2 mittleres, 3 (starkes)
Fehlverhalten' , `nameOfBehaviour` VARCHAR(20) NOT NULL COMMENT
'PK für Behaviour, Verhalten des vSoS, hier: Lehrer anschauen' ,
`seatplace` VARCHAR(3) NOT NULL , PRIMARY KEY
(`nameOfBehaviour`)) ENGINE = InnoDB;
```

Abbildung 11: Beispiel SQL-Befehle von phpMyAdmin für Erstellung der Attribute für "behaviour", ohne phpMyAdmin müsste dieser Befehl manuell in die Kommandozeile eingetragen werden

Abbildung 12: Beispiel Erstellen einer Beziehung zwischen "behaviour" und "virtualStudent" mittels des PK bzw. FK "seatplace"

10 Anhang

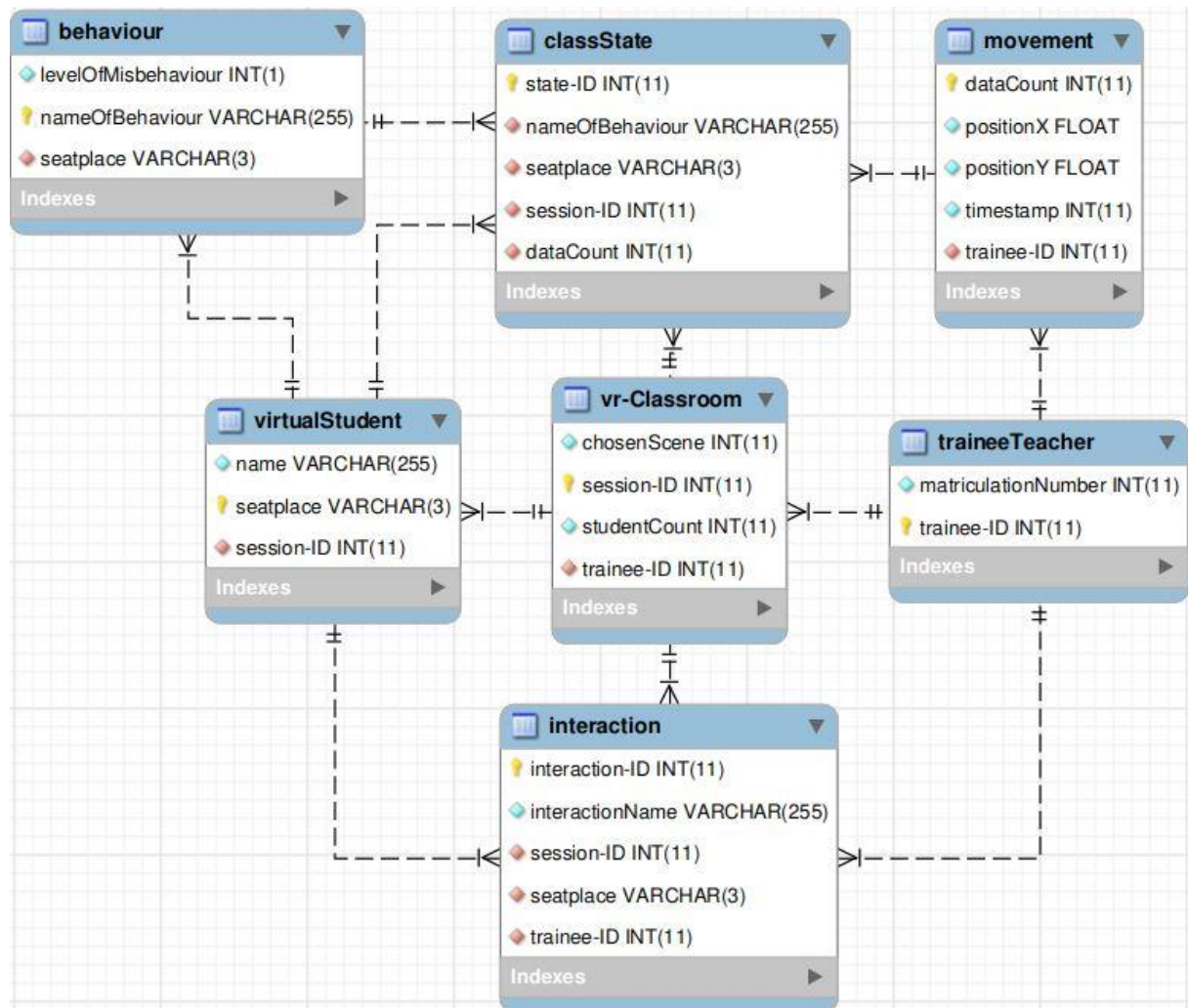


Abbildung 13: ERM in MySQL Workbench, Beziehungen anders als bei Chen-Notation, durch Auswahl eines PK werden alle FK farblich hervorgehoben

```
SELECT
    t.matriculationNumber,
    v.studentCount,
    v.`session-ID`
FROM
    traineeTeacher t,
    `vr-Classroom` v
WHERE
    t.matriculationNumber = "689576" AND t.matriculationNumber = v.matriculationNumber
```

Abbildung 14: Abfrage 1

```
SELECT
    COUNT(b.levelOfMisbehaviour) "misbehaviourCount"
FROM
    behaviour b,
    traineeTeacher t,
    `vr-Classroom` v
WHERE
    t.matriculationNumber = "689576" AND t.matriculationNumber = v.matriculationNumber
```

Abbildung 15: Abfrage 2, mit „COUNT“ Funktion

10 Anhang

```
#Prozedur getDistance
DELIMITER $$
CREATE DEFINER=`phpmyadmin`@`localhost` PROCEDURE `getDistance`()
BEGIN
    DECLARE iterator INT DEFAULT 0;
    DECLARE i INT DEFAULT 0;
    DECLARE x1 FLOAT;
    DECLARE x2 FLOAT;
    DECLARE y1 FLOAT;
    DECLARE y2 FLOAT;
    DECLARE distance FLOAT DEFAULT 0;
    DECLARE inc INT DEFAULT 0;

    SELECT COUNT(*) FROM movement into iterator;

    WHILE i < iterator DO
        SELECT positionX into x1 from movement LIMIT i, 1;
        SELECT positionY into y1 from movement LIMIT i, 1;
        SET inc = i + 1;
        SELECT positionX into x2 from movement LIMIT inc, 1;
        SELECT positionY into y2 from movement LIMIT inc, 1;
        SET distance = distance + calcDistance(x1,x2,y1,y2);
        SET i = i+1;
    END WHILE;
    SELECT distance "walkedDistance";
END$$
DELIMITER ;
```

Abbildung 16: Für Abfrage 3: Prozedur "getDistance"

```
#Funktion calcDistance
DELIMITER $$
CREATE DEFINER=`phpmyadmin`@`localhost` FUNCTION `calcDistance`(`x1` FLOAT, `x2` FLOAT, `y1` FLOAT, `y2` FLOAT)
RETURNS float
BEGIN
    DECLARE dist FLOAT;

    SET dist = SQRT(POW(x2 - x1,2) + POW(y2 - y1,2));

    RETURN dist;
END$$
DELIMITER ;
```

Abbildung 17: Für Abfrage 3: Funktion "calcDistance"